UCRL-CR-148022
B525176

Multi-Programmatic and Institutional Computing Capacity Resource Attachment 2 Statement of Work

M. Seager

April 15, 2002

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

Approved for public release; further dissemination unlimited

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

University of California Lawrence Livermore National Laboratory

Multi-Programmatic and Institutional Computing Capacity Resource

Attachment 2
Statement of Work

B525176

Version 5
April 15, 2002

Science in the National Interest

Lawrence Livermore National Laboratory

Department of Energy
University of California

Lawrence Livermore National Laboratory ensures national security and applies science and technology to important problems of our time.

TABLE OF CONTENTS

1 Background

1.1 Multi-programmatic and Institutional Computing (M&IC)

Lawrence Livermore National Laboratory (LLNL) has identified high-performance computing as a critical competency necessary to meet the goals of LLNL's scientific and engineering programs. Leadership in scientific computing demands the availability of a stable, powerful, well-balanced computational infrastructure, and it requires research directed at advanced architectures, enabling numerical methods and computer science.

To encourage all programs to benefit from the huge investment being made by the Advanced Simulation and Computing Program (ASCI) at LLNL, and to provide a mechanism to facilitate multi-programmatic leveraging of resources and access to high-performance equipment by researchers, M&IC was created.

The Livermore Computing (LC) Center, a part of the Computations Directorate Integrated Computing and Communications (ICC) Department can be viewed as composed of two facilities, one open and one secure. This acquisition is focused on the M&IC resources in the Open Computing Facility (OCF).

For the M&IC program, recent efforts and expenditures have focused on enhancing capacity and stabilizing the TeraCluster 2000 (TC2K) resource. *Capacity* is a measure of the ability to process a varied workload from many scientists simultaneously. *Capability* represents the ability to deliver a very large system to run scientific calculations at large scale.

In this procurement action, we intend to significantly increase the capability of the M&IC resource to address multiple teraFLOP/s problems, and well as increasing the capacity to do many 100 gigaFLOP/s calculations.

1.2 Partnership with the Stockpile Stewardship Program (SSP)

The M&IC platforms form part of the unclassified computing environment at LLNL. This environment is called the Open Computing Facility (OCF). Some of the platforms in the OCF represent primarily Stockpile Stewardship Program investments (for instance, the unclassified ASCI systems). Others, such as the SMP clusters, represent fairly evenly divided investments between multiple programs (including the SSP) and the institution.

The support infrastructure (everything but the compute platforms) is covered both by M&IC and by the NNSA Stockpile Stewardship Program (SSP), and the partners share the resources. For instance, the IBM HPSS storage environment represents an ~80/20 split, with the SSP making the heavier investment. On the other hand, the NFS home space environment was procured by M&IC. The visualization environment is more heavily funded by the SSP, but the System Area Network is supported by the M&IC. The LC balances these investments according to the utilization of the broad support infrastructure by the partners. The end result is a more powerful Open Computing Facility than any program (or the Institution) could afford alone.

1.3 Current M&IC Platforms

Over the last five years, M&IC has acquired several computational platforms: the GPS, TC98, LX and TC2K clusters, and Sunbert. In addition, M&IC has made co-investments in unclassified ASCI systems in return for a fractional allocation of these platforms: ASCI Frost (~20% M&IC allocation), and ASCI Blue (~5-10% M&IC allocation). The following information summarizes the existing generation of M&IC platforms and the unclassified ASCI platforms in which M&IC has an allocation. A more complete description of the systems can be found at URL http://www.llnl.gov/icc/lc/mic/mic.html.

1.3.1 TeraCluster (TC98)

The TeraCluster consists of 24 Compaq AlphaServer 4100 nodes (4-way 553MHz Alpha EV5.6, 1GiB of memory, 60GB local disk) and 8 Compaq DS20 nodes (2-way 500-MHz Alpha EV6, 1GiB of memory, 8GB local disk). The nodes communicate via 100BaseT Ethernet. All nodes run Compaq Tru64 Unix as the OS. TC98 has a total computing capacity of about 118 gigaFLOP/s. A more complete description of the system can be found at the URL: http://www.llnl.gov/icc/lc/OCF_resources.html#tera_cluster

1.3.2 TeraCluster 2000 (TC2K)

TC2K is a follow-on to the TeraCluster (TC98). TC2K is a serial number 1 Compaq Alpha SC system that was delivered in September 1999. TC2K has the same hardware and software architecture of the LANL Q system although ES40s are the node "building blocks" of TC2K rather than the ES45s in Q. TC2K is composed of 128 Compaq AlphaServer ES40 nodes (4-way 667MHz Alpha EV6.7), configured as 116 compute nodes with 2 GiB of memory, and two login nodes with 8 GiB of memory. This platform has approximately 7 TB of disk configured to form a parallel file system using Compaq's CFS (Cluster File System) software. The nodes communicate via a 128-way Quadrics QsNet Elan3 switch. TC2K has a theoretical peak performance of 681 gigaFLOP/s. A more complete description of the system can be found at the URL: http://www.llnl.gov/icc/lc/OCF_resources.html#tc2k

1.3.3 LX Cluster

The LX Cluster consists of 32 Compaq DS20E nodes (2-way 667MHz Alpha EV6.7, 2GiB of memory), with 7GB of local disk. The nodes communicate via 100BaseT Ethernet. All nodes run Red Hat Linux 7.1 as the OS. The LX Cluster has a total computing capacity of about 85 gigaFLOP/s. A more complete description of the system can be found at the URL: http://www.llnl.gov/icc/lc/OCF_resources.html#linux

1.3.4 GPS Cluster

The GPS Cluster was acquired in FY01 to significantly increase the capacity environment of the OCF. The GPS Cluster is composed of 16 Compaq ES45 nodes (4-way 1GHz Alpha EV6.8, with a mix of 4-32GiB memories) and one Compaq GS320 node (32-way 1GHz Alpha EV6.8, with 32GiB memory). The nodes communicate via 100BaseT Ethernet. All nodes run Compaq Tru64 Unix as the OS. The GPS Cluster has a total compute capacity of about 192 gigaFLOP/s. A more complete description of the system can be found at the URL: http://www.llnl.gov/icc/lc/OCF_resources.html#gps

1.3.5 Sunbert

Sunbert is a Sun Enterprise 6000 system, with 24 250MHz UltraSPARC-II cpus, 16GiB of memory, and about 500GB of local disk. Sunbert runs Sun Solaris OS. Sunbert has a theoretical peak speed of about 12 gigaFLOP/s. A more complete description of the system can be found at the URL: http://www.llnl.gov/icc/lc/OCF_resources.html#Sunbert

1.3.6 ASCI Frost

Because of institutional investment, a portion of ASCI Frost is allocated to M&IC use. ASCI Frost is composed of 68 IBM SP NightHawk-2 nodes (16-way 375MHz PowerPC3, 16GiB of memory), with a total of 5TB of local disk, and 20TB of disk configured as a parallel file system utilizing IBM's GPFS (General Parallel File System). The nodes communicate with each other over IBM's Dual Plane Colony Double-Single switches, and runs IBM's AIX Unix OS and PSSP software. ASCI Frost has a theoretical peak speed of about 1.632 teraFLOP/s. A more complete description of the system can be found at the URL: http://www.llnl.gov/icc/lc/OCF_resources.html#frost

1.3.7 ASCI Blue-Pacific (CTR)

Because of institutional investment, a portion of ASCI Blue-Pacific is allocated to M&IC use. ASCI Blue-Pacific is composed of 280 IBM SP Silver nodes (4-way 332MHz PowerPC 604e, 1.5GiB), with 3TB of local disk, and 16TB of disk configured as a parallel file system utilizing IBM's GPFS (General Parallel File System). The nodes communicate with each other over IBM's TB3MX switch, and runs IBM's AIX Unix OS and PSSP software. ASCI Blue-Pacific has a theoretical peak speed of about 728 gigaFLOP/s. A more complete description of the system can be found at the URL: http://www.llnl.gov/icc/lc/OCF_resources.html#ibm

1.4 M&IC Applications Overview

LLNL is in the forefront of the evolution toward effective and practical computational science in all its forms. To continue this role, we must continue to provide the computational tools that a wide user community needs to advance their scientific research.

LC has recently conducted a detailed analysis of M&IC computing requirements, finding that our users support projects that address a wide range of important scientific issues and pressing national and international concerns. The total computing needs of these projects far exceed the current M&IC capability and capacity. The pace of progress on many of these projects is being paced by the available computing resources, and access to additional computing cycles will result in faster progress. As shown in Table 1.5-1, these projects span the technical directorates and support the major programs of the Laboratory. The technical directorators noted in this table are Biology & Biotechnology Research Program (BBRP), Chemistry & Materials Science (C&MS), Computation (Comp), Defense & Nuclear Technologies (DNT), Energy & Environment Directorate (EED), Engineering (Eng), National Ignition Facility Programs (NIF), Nonproliferation, Arms Control & International Security (NAI), and Physics and Advanced Technologies (PAT).

Computation is now a mainstream method in theoretical science at LLNL – essential when highly simplified but analytically intractable models are explored or complex multiphysics phenomena must be understood quantitatively. As we understand more and more truly basic science, the Laboratory is looking to computation to make the vital numerical connections among disparate models that constitute the foundation of both pure and appliced science.

		Technical Directorates Involved in this Project								
	Project ID	BBRP	C&MS	Comp	DNT	EED	Eng	NIF	NAI	PAT
1	ALPS			□	▨					▨
2	DJEHUTY			□	□					□
3	AMRh				□					
4	Fermion MC			□	□					□
5	DD-ICF				□					
6	Z3				□					
7	Mat-Shock		□		▨		▨	▨		▨
8	Mat-Rad		□				▨			▨
9	Cell Modeling		□							
10	Biochem		□							
11	CompBio	□	□	▨						▨
12	GFMD		▨							□
13	BOUT									□
14	NuclStruct									□
15	JEEP	▨		□					▨	□
16	PHENIX/HBT		▨				▨			□
17	MD3D			▨						□
18	pF3d				□			▨		□
19	NIF gas						□	▨		
20	EIGER codes						□			
21	NDE						□			
22	E3D			□		□	□	▨	▨	▨
23	HP-CFD					□			▨	
24	NUFT-C					□				
25	HR-GCS			▨		□				
26	AtmosChem					□				
27	Earthquake						□			

Figure 1.4-1 A recent analysis of M&IC computing requirements included projects from all nine technical directorates at the Laboratory.

An enormous role is also being played in experimental science by M&IC projects. Many of these projects have achieved such sophistication that direct comparisons between full-scale experiment and simulations based on ab initio models are now possible. But it is often the case that these direct comparisons are limited by the available computing resources. As seen in Table 1.5-2, the vast majority of M&IC projects are closely tied to experiment, and nearly half provide some support to NIF, the National Ignition Facility, a major program at the Laboratory. Many projects have an even stronger experimental tie, being used to design experiments or facilities.

Project ID		Connect to exp?	Connect to NIF?	Exp design?	Facility design?
1	ALPS	yes	yes		
2	DJEHUTY				
3	AMRh	yes	yes		
4	Fermion MC				
5	DD-ICF	yes	yes		
6	Z3	yes	yes		
7	Mat-Shock	yes	yes		
8	Mat-Rad	yes	yes		
9	Cell Modeling				
10	FP-Biochem	yes			
11	CompBio	yes		yes	
12	GFMD	yes			
13	BOUT	yes			
14	NuclStruct				
15	JEEP	yes	yes		
16	PHENIX/HBT	yes			
17	MD3D	yes			
18	pF3d	yes	yes	yes	yes
19	NIF gas	yes	yes		yes
20	EIGER	yes			
21	NDE	yes	yes		
22	E3D	yes	yes	yes	yes
23	HP-CFD	yes	yes	yes	
24	NUFT-C	yes		yes	
25	HR-GCS	yes			

Figure 1.4-2 These M&IC computationally intensive projects have close connection with experiment.

M&IC codes fall in a broad range of applications spanning the physical, environmental and biological sciences. Typically these codes are complex time-dependent simulations of multiple physical processes, where the processes are often tightly coupled and will require physics models linking microscale phenomena to macroscopic response. Generally these simulations are multidimensional with the trend toward full three-dimensional treatment of physical space. Table 1.5-3 illustrates that many of these applications fall into the following broad classifications:

- molecular dynamics simulations (MD)
- adaptive-mesh-refinement codes (AMR)
- laser-plasma-interaction simulations (LPI)
- computational-fluid-dynamics simulations (CPD)
- hydrodynamic simulations (hydro)
- coupled radiation-transport and hydrodynamic simulations (rad-hydro)
- radiation-transport simulations (rad trans)

Project ID		MD	AMR	LPI	CFD	hydro	rad-hydro	rad trans
1	ALPS		x	x		x	x	
2	DJEHUTY					x	x	
3	AMRh		x			x	x	
4	Fermion MC							
5	DD-ICF			x		x	x	
6	Z3			x				
7	Mat-Shock	x						
8	Mat-Rad	x						
9	Cell Modeling							
10	FP-Biochem	x						
11	CompBio	x						
12	GFMD	x						
13	BOUT					x		
14	NuclStruct							
15	JEEP	x						
16	PHENIX/HBT							
17	MD3D	x						
18	pF3d			x				
19	NIF gas					x		
20	EIGER							
21	NDE							
22	E3D							
23	HP-CFD				x			
24	NUFT-C							
25	HR-GCS				x			x

Figure 1.4-3 : Many of these M&IC applications share common algorithmic approaches, while differing markedly in the details of the implementations.

Demanding national security issues are driving intense development of M&IC applications codes. Advanced numerical algorithms require innovative software techniques to achieve the necessary delivered performance on advanced computing platforms. Current application characteristics and expected trends are described below, although M&IC applications codes and numerical algorithms will continue to evolve rapidly.

The following are some of the major M&IC code characteristics essential to an understanding of the vision of an ideal computing environment.

The codes often model multiple types of physical/chemical/biological/environmental processes, generally in a single (usually monolithic) application, in a time-evolving manner with direct coupling between all simulated processes. They do so using a variety of computational methods, often through a separation or "split" of the various processes and coupling terms. This process often involves solving first one type of model, then the next, then another, and then repeating this sequence for every time step. Some algorithms are categorized as explicit in time while others are fully implicit or semi-implicit and typically involve iterative solvers of some form. Some special wavefront "sweeps" are employed for specific direct-solve algorithms. Numerous research efforts are actively exploring novel and alternative methods and algorithms for possible application to problems of interest.

The calculations are of various sizes, with some treating millions of particles or spatial zones (cells), with an expected requirement for many applications to get to the point of using upwards of a billion or more particles/cells. The equations are typically solved by spatial discretization. Discretization over energy and/or angle, in addition, can increase the data space size by 10 to 1000 times. In the final analysis, thousands of variables will be associated with each zone. Monte Carlo algorithms will treat millions to billions of particles distributed throughout the problem domain. The parallelization strategy for many codes is based upon decomposition into spatial domains. For some applications, codes will use decomposition over angular or energy domains, as well.

Currently, almost all codes use the standard message passing interface (MPI) for parallel communication, even between processes running on the same SMP. In addition, some applications utilize OpenMP for SMP parallelism. The efficiency of OpenMP SMP parallelism depends highly on the underlying compiler implementation (i.e., the algorithms are highly sensitive to OpenMP overheads). Also, it is possible in the future that different physics models within the same application might use different communication models. For example, an MPI-only main program may call a module that uses the same number of MPI processes, but also uses threads (either explicitly or through OpenMP). In the ideal system, these models should interoperate as seamlessly as possible. Mixing such models mandates thread-safe MPI libraries. Alternative strategies may involve calling MPI from multiple threads with the expectation of increased parallelism in the communications; such use implies multi-threaded MPI implementations as well.

Current codes are based on a single program multiple data (SPMD) approach to parallel computing. However, director/worker constructs are often used. Typically, data are decomposed and distributed across the system and the same execution image is started on all MPI processes and/or threads. Exchanges of remote data occur for the most part at regular points in the execution, and all processes/threads participate (or just pretend to) in each such exchange. Data are actually exchanged with individual MPI send-receive requests, but the exchange as a whole can be thought of as a "some-to-some" operation with the actual data transfer needs determined from the decomposition. Weak synchronization naturally occurs in this case because of these exchanges, while stronger synchronization occurs because of global operations, such as reductions and broadcasts (e.g., MPI_allreduce), which are critical parts of iterative methods. It is quite possible that future applications will use functional parallelism, but mostly in conjunction with the SPMD model. Parallel input-output (I/O) and visualization are areas that may use such an approach with functional parallelism at a high level to separate them from the physics simulation, yet maintain the SPMD parallelism within each subset. There is some interest in having visualization tools dynamically attach to running codes and then detach for interactive interrogation of simulation progress. Such mixed approaches are also under consideration for some physics models.

Many applications use unstructured spatial meshes. Even codes with regular structured meshes may have unstructured data if they use cell-by-cell continuous adaptive mesh refinement (AMR). In an unstructured mesh, the neighbor of zone (i) is not zone (i+1), and one must use indirection or data pointers to define connectivity. Indirection has been implemented in several codes through libraries of gather-scatter functions that handle both on-processor as well as remote communication to access that neighbor information. The communication support is currently built on top of MPI and/or shared memory to get that neighbor information. These scatter-gather libraries are two-phased for good efficiency. In phase one, the gather-scatter pattern is presented and all local memory and remote memory and communications structures are initialized. Then in phase two, the actual requests for data are made, usually many, many times. Thus, the patterns are extensively reused over and over again. Also, several patterns will coexist simultaneously during a timestep for various data. Techniques like AMR and reconnecting meshes can lead to pattern changes at fixed points in time, possibly every cycle or maybe only after several cycles.

Memory for arrays and/or data structures is typically allocated dynamically, avoiding the need to recompile with changed parameters for each simulation size. This allocation requires compilers, debuggers, and other tools that recognize and support such features as dynamic arrays and data structures, as well as memory allocation intrinsics and pointers in the various languages.

Many of the physics modules will have low compute–communications ratios. It is not always possible to hide latency through non-blocking asynchronous communication, as the data are usually needed to proceed with the calculation. Thus, a low-latency communications system is crucial.

Many of the physics models are memory intensive, and will perform only about 1 FLOP per load from memory. Thus, performance of the memory sub-system is crucial, as are compilers that optimize in terms of cache blocking, loop unrolling-rolling, loop nest analysis, etc. Many codes have loops over all points in an entire spatial decomposition domain. This coding style is preferred by many for ease of implementation and readability of the physics and algorithms. Although recognized as problematic, effective automatic optimization is preferred, where possible.

The multiple physics models embedded in a large application may have dramatically varying communication characteristics, i.e., one model may be bandwidth-sensitive, while another may be latency-sensitive. Even the communications characteristics of a single physics model may vary greatly during the course of a calculation as the spatial mesh evolves or different physical regimes are reached and the modeling requirements change. In the ideal system, the communications system should handle this disparity without requiring user tuning or intervention.

Although static domain decomposition is used for load balancing as much as possible, there are also definite needs for dynamic load balancing, in which the work is moved from one processor to another. One obvious example is for codes using AMR methods, where additional cells may be added or removed during the execution wherever necessary in the mesh. It is also expected that different physical processes will be regionally constrained and, as such, will lead to load imbalances that can change with time as different processes become "active" or more difficult to model. Any such dynamic load balancing is expected to be accomplished through associated data migration explicitly done by the application itself. This re-balancing might occur inside a time step, every few timesteps, or infrequently, depending on the nature of the problem being run. In the future, code execution may also spawn and/or delete processes to account for the increase and/or decrease in the total amount of work the code is doing at that time.

1.5 M&IC Scientific Software Development Environment

The following are some of the major characteristics of the software development environment in an ideal scenario.

A high degree of code portability and longevity is a major objective. Many M&IC codes must execute at multiple sites. Development, testing and validation of 3D, full-physics, full system applications requires four to six years. The productive lifespan of these codes is at least ten years. Thus these applications must span not only today's platforms but any possible future system. Codes will be developed in standards-conforming languages so non-standard compiler features are of little interest unless they can be made transparent. The use of Cray Pointers in Fortran is an exception to our reliance on standard features. We also will not take advantage of any idiosyncratic features of optimization, unless they can be hidden from the codes (e.g., in a standard library). Non-standard "hand tuning" of codes for specific platforms is antithetical to this concept.

A high-performance, low-latency MPI environment that is robust and scalable is crucial to us. Today applications are utilizing all the features of MPI 1.1 functionality. Many features of MPI-2 functionality are also in current use. Hence, a full, robust and efficient implementation of MPI-2 is of tremendous interest. A POSIX compliant-thread environment is also crucial and a Fortran95-threads interface is also important. All libraries need to be thread-safe. MPI should be multi-threaded as well as thread-safe. We should not have to tune the MPI-runtime environment for different codes and different problem sizes. In our estimation, bandwidth of 0.2 bytes/second/peak OP/second/SMP and an end-to-end MPI ping-pong latency of less than 10 microseconds or better will provide the desired performance. Since we are talking about systems with tens of thousands of processors, it is vitally important that the MPI implementation scale to the full size of the system. This scaling is both in terms of efficiency (particularly of the MPI_ALLREDUCE functionality) as well as the efficient use of buffer memory. M&IC applications are carefully programmed so that MPI RECIEVE operations are posted before the corresponding SEND operation. This allows for minimal (and hence scalable) MPI buffer space allocations.

M&IC applications require the ability for each MPI task to access up to 2.0GB of physical memory. The large memory sizes of MPI tasks requires that nodes be configured with 1-2 GiB of real memory per processor.

We will expect the compilers to do the vast majority of code optimization through simple easy-to-use compiler switches (e.g. -On). Also, we will expect the compilers to have options to range check arrays under debug mode, as well as a way to trap underflow, overflow, divide by zero, etc. Parallelization through the OpenMP specifications is of particular interest and is expected for Fortran95, C, and C++. OpenMP parallelization must function correctly in programs that also use MPI. OpenMP Version 2.0 support for Fortran95, Version 1.0 for C/C++ is highly favored, while automatic parallelization is of some interest, if it is efficient and does not drive compile times to unreasonable lengths. Any information the compiler can provide about the optimizations it performed is useful. Compiler parallelism has to work in conjunction with MPI. All compilers must be fully ANSI-compliant.

The availability of standard, platform-independent tools is necessary for a portable and powerful development environment. Examples of these tools are GNU software (especially GNU make, but others as well), TotalView debugger (the current debugger on all M&IC platforms), dependency builders (Fortran USE & INCLUDE as well as #include), preprocessors (CPP, M4), source analyzers (lint, flint, etc), hardware counter libraries and communications profilers (VAMPIR, etc). Tools that work with a source code should fully support the most current language standards. A standard API to give debuggers and performance analyzers access to the state of a running code will allow us to develop our own tools or to use a variety of tools developed by others. The Distributed Process Class Library (DPCL) is an emerging public domain API that meets this need. These performance and debugging tools must not require privileged access modes, such as root user nor compromise the security of the runtime environment.

We must have parallel debuggers that allow us to debug parallel applications within an SMP or node and that permit parallel application debugging applications utilizing multiple nodes or SMPs. This includes MPI-only as well as mixed MPI + threads and/or OpenMP codes. In the best of all possible worlds, the debugger will allow effective debugging of jobs using every CPU on the system. Practical use of large fractions of the machine by an application under the control of the debugger requires that the debugger be highly integrated into the system initiated parallel checkpoint/restart and GANG scheduling mechanisms. Some specific features of interest include the following:

- breakpoints,
- fast conditional breakpoints,
- fast conditional watchpoints on memory locations,
- a save-restore state for backing up via checkpoint/restart mechanism,
- complex selections for data display (possibly even programming support with loops, conditionals, local variables, etc),
- support for array statistics (min, max, etc),
- attaching/detaching to running jobs,
- an initialization file that knows where the sources are and what options we want etc., and
- a command-line interface in addition to a GUI (e.g. for working over slow phone lines from home).

The capability to visually examine slices and subsets of multidimensional arrays is a feature that has proven useful. The debugger should allow complex selections for data display to be expressible with Fortran95 and C language constructs and features. It should support applications written in a mixture of the baseline languages (Fortran95, C and C++), support Cray-style pointers in Fortran77, and be able to dive on templated functions and handle complex template evaluation capabilities in C++. It should be able to debug compiler-optimized code since problems sometimes go away at debug levels, although less symbolic and contextual information will be available to the debugger at higher levels of optimization. Our build environment involves accessing source code from NFS mounted file systems with likely compiling and linking of the executable in alternate directories. This process may have implications, depending on how the compiler tells the debugger to find the source code. The debugger currently used in the Tri-Laboratory ASCI PSE CDE is the TotalView debugger from Etnus (see URL: http://www.etnus.com/Products/TotalView/index.html).

Because most M&IC codes are memory-access intensive, optimizing the spatial and temporal locality of memory accesses is crucial for all levels of the memory hierarchy. To tune memory distribution in a NUMA machine, it is necessary to be able to specify where memory is allocated. To optimally use memory and to reuse data in cache, it is also necessary to cause threads to execute on CPUs that quickly access particular NUMA regions and particular caches. Expressing such affinities should be an unprivileged operation. Threads generated by a parallelizing compiler (OpenMP or otherwise) should be aware of memory-thread affinity issues as well.

Other ramifications of the large memory footprint of M&IC codes is that they require large delivered memory bandwidth as seen by the applications actual memory reference patterns. This is a requirement that stresses the memory subsystem when the applications display regular memory reference patterns and have a high degree of cache utilization and high degree of cache line utilization for application memory reference payload delivery. In addition, because many of these memory-access intensive codes have random memory access patterns (due to indirect addressing or complex C++ structure and method dereferencing brought about from implementing discretization of spatial variables on block structured or unstructured grids) and hence access thousands to millions of standard UNIX™ 4KB VM pages every timestep, "large page support" in the operating system for efficient utilization of the microprocessor virtual to real memory translation functionality and caches is required for efficient use of the hardware. This is due to the fact that hardware TLBs have a limited number of entries (although caching additional entries in L1 cache helps but does not solve the problem) and having, say, 256 MiB page size will significantly reduce the number of TLB entries required for large memory-access M&IC code VM to real memory translations. Since TLB misses (that are not cached in L1) are very expensive, this feature can significantly enhance M&IC application performance.

Many of our codes could benefit from a high-performance, standards-conforming, parallel I/O library, such as MPI-I/O. Many M&IC applications development teams now consider the ability to do MPI-2 dynamic tasking an essential item for future M&IC code development efforts. In addition, low latency GET/PUT operations for transmission of single cache lines is viewed as essential for domain overloading on a single SMP or node. However, many implementations of the MPI-2 MPI_GET/MPI_PUT mechanisms do not have lower latency than MPI_SEND/MPI_REC, but do allow for multiple outstanding MPI_GET/MPI_PUT operations to be active at a time. This approach although appealing to MPI-2 library developers, put the onus of latency hiding on the applications developer, who will rather think about physics issues. Future M&IC applications require a very low latency (as close to the SMP memory copy hardware latency as possible) for GET/PUT operations.

It is advantageous to have support for translating big-endian, little-endian, and Cray Research PVP data representations to the system's native data forms. Especially useful will be automatic I/O filters on a file-by-file basis that will do this at read-write time.

Effectively tuning an application's performance requires detailed information on its timing and computation activities. On an SMP or node, a timer that is consistent between threads or tasks running on different CPUs in that same SMP or node is useful. Frequent use of the timer implies high-resolution (10 microseconds or better) and low overhead. In addition, other hardware performance monitoring information such as the number of cache misses, TLB misses, floating-point operations, etc. can be very helpful. All modern microprocessors contain counters that gather this kind of information. The data in these counters can be made available separately for each thread or process through tools or programming libraries accessible to the user. For portability, our tools are targeting the PAPI library for hardware counters (`http://icl.cs.utk.edu/projects/papi/`). To limit instrumentation overhead, a version of their tools that support multiplexing of hardware counters, and sampling of instructions in the pipeline is easier to use. Note that this facility requires that the operating system context switch these counters at process or heavy weight (OS scheduled) thread level and that the POSIX or OpenMP runtime libraries context switch the counters on light weight (library scheduled) thread level. Furthermore, these counters can be available to users that do not have privileged access, such as the root user. Per-thread OS statistics must be available to all users via a command line utility as well as a system call. One example of such a feature is the `kstat` facility: a general-purpose mechanism for providing kernel statistics to users. Both hardware and counter statistics must provide virtualized information, so that users can make the correct attribution of performance data to application behaviors.

We need to have early access to new versions of system and development software, as well as other supplied software. Software releases of the various products should be synchronized with operating system releases to ensure compatibility and interoperability.

1.6 M&IC Applications Execution Environment

The following are some major characteristics of the M&IC ultra-scale applications execution environment.

It is crucial to be able to run a single parallel job on the full system using all resources available for a week or more at a time. This is called a "full-system run." Any form of interruption should be minimized. The capability for the system and application to "fail gracefully" and then recover quickly and easily is an extremely important issue for such calculations. We also expect to be running a large number of jobs on thousands of processors each for hundreds of hours. These will require significant system resources, but not the entire system. The capability of the system to "fail gracefully," so that a failure in one section of the system will only affect jobs running on that specific section, is important. From the applications perspective, the probability of failure should be proportional to the fraction of the system utilized. A failed section should be repairable without bringing down the entire system.

A single simulation may run over a period of months as separate restarted jobs in increments of days running on varying numbers of processors with different physics models activated. Output files produced by a code on one set of processors need to be efficiently accessible by another set of processors, or possibly even by a different number of processors, to restart the simulation. Thus an efficient cluster wide file system is essential. Ideally, file input and output between runs should be insensitive to the number of processors before and after a restart. It should be possible to restart a job across a larger or smaller number of processors than originally used, with only a slight difference in performance visible.

M&IC applications write many restart and visualization dumps during the course of a run. A single restart dump will be about the same size as the job's memory image, while visualization dumps will be perhaps from 1 to 10 % of that size. Restart dumps will typically be scheduled based on wall clock periods, while visualization dumps are scheduled entirely on the basis of internal physics simulation time. We usually create visualization dumps more frequently than restart dumps. System reliability will have a direct effect on the frequency of restart dumps; the less reliable the system is, the more frequently restart dumps will be made and the more sensitive we will be to I /O performance. We have observed on previous generation M&IC platforms that restart dumps comprise over 75% of the data written to disk. Most of this I/O is wasted in the sense that restart dumps are overwritten as the simulation progresses. However, this I/O must be done so that the simulation is not lost to a platform failure. This leads us to the notion that cluster wide file system (CWFS) I/O can be segregated into two portions: productive I/O and defensive I/O. Productive I/O is the writing of data that the user needs to do science (visualization dumps, traces of key physics variables over time, etc.). Defensive I/O is done to manage a large simulation run over a period of time much larger than the platform MTBF. Thus, one will like to minimize the amount of resources devoted to defensive I/O and computation lost due to platform failure. This can be accomplished by procuring resources with a high MTBF.

Operationally, applications teams push the large restart and visualization dumps (already described) off to HPSS tertiary storage within the wall clock time between making these dumps. The disk space mentioned elsewhere in this document is insufficient to handle M&IC applications long-term storage needs. HPSS is the archive storage system of M&IC and compatibility with it is needed. Thus, a highly usable mechanism is required for the parallel high-speed transport of 1's to 10's of TB of data from the CWFS to HPSS.

We need a resource manager-job scheduler that deals with all aspects of the system's resources, not with just the processors and the time allocations. Factors that should be considered include processors, processes, memory, interconnects, disks, visualization engines, etc. It will be essential for this resource manager-scheduler to handle both batch and interactive execution of both serial and parallel programs (MPI and threaded) from a single processor to the full cluster. The manager-scheduler will provide a way to implement policies on selecting and executing various problems (problem size, problem run time, timeslots, preemption, users' allocated share of machine, etc). Also, a method will be provided for users to connect to executing batch jobs to query or change problem status or parameters. The tool(s) will schedule jobs to provide for process-to-processor affinity. We are currently using LLNL's DPCS on the ASCI Blue Pacific and White systems as well as existing Linux clusters and other M&IC resources.

Our codes and users will benefit from a robust, globally visible, high-performance, parallel file system. It is essential that all file systems have large file (64b file pointer) offsets. A 32b file pointer is clearly insufficient.

1.7 M&IC Operational Environment

LC operates our production systems 24 hours per day, 7 days per week, including holidays. The prime shift is from 8 AM to 5 PM, Pacific Time Zone. LLNL local users access these systems via the 1 Gigabit Ethernet local-area network (LAN). MCR will operate in this environment.

The prime shift period will be devoted primarily to interactive applications development, interactive visualization, relatively short time-limit, large CPU count (e.g., over half the system CPUs), high priority production runs and extremely long running, smaller CPU count (e.g, 64-512), lower priority production runs. Night shifts, as well as the weekend and holiday periods, will be devoted to extremely long-running jobs. Checkpointing and restarting jobs will take place as necessary to schedule this heterogeneous mix of jobs under dynamic load and job priorities on

MCR. Because the workload is so varied and the demands for CPU time oversubscribe the machine by several factors, resource scheduling is an essential production requirement. In addition to application-initiated checkpoint/restart, M&IC applications have the ability to do application based restart dumps. These interim dumps, as well as visualization output, will be stored on HPSS-based archival systems or sent to the VIEWS visualization corridors via the system-area network (SAN) and external "Jumbo Frame" 1 Gigabit Ethernet interfaces. Depending upon system protocol support, IP version 4 and lightweight memory-to-memory protocol (e.g., iWARP) traffic will be carried in this environment.

A single point of system administration will allow the configuration of the entire system from a common server. The single server will control all aspects of system administration in aggregate. Examples of system administration functions include modifying configuration files, editing mail lists, software, upgrades and patch (bug fix) installs, kernel parameter changes, file system-disk manipulation, reboots, user account activities (adding, removing, modifying), performance analysis, hardware changes, and network changes. A hardware and software configuration management system that profiles the system hardware and software configuration as a function of time and keeps track of who makes changes is essential.

The ability to dynamically monitor system functioning in real time and allow system administrators to quickly diagnose hardware, software (e.g., job scheduler) and workload problems and take corrective action is also essential. These monitoring tools must be fast, scalable and display data in a hierarchal schema. The overhead of system monitoring and control will necessarily need to be low in order to not destroy large job scalability (performance).

LLNL's Distributed Production Control System (DPCS) will manage the queue of pending batch jobs, deciding when to initiate pending jobs so as to achieve LLNL management objectives. DPCS is a mature system that has been managing LLNL's supercomputer workloads since 1992. DPCS requires an underlying resource manager to allocate nodes in the cluster for jobs being initiated, initiate the required tasks, and establish the switch interconnects between these tasks. We intend to utilize the Quadrics Resource Management System (RMS) in this role initially. Our intent is to replace RMS with a more scalable, open-source resource manager presently under development at LLNL: Simple Linux Utility for Resource Management (SLURM). SLURM will support initiating and managing MPI jobs utilizing QsNet and should be ready for deployment in late 2002.

The operating environment will conform to DOE security requirements for Unclassified systems. Software modifications will be made in a timely manner to meet changes to those security requirements.

1.8 Utilization of Existing Facilities

An existing facility, the main computer floor of B439, will be used for siting the MCR system. This facility has approximately 8,000-9,000 ft^2 and 1.9 MW of power for the computing system and peripherals and associated cooling available for this purpose. Facilities modifications to provide power hook-up and cooling will be required to site the MCR system. Thus, it is essential the Offeror make available to the University detailed and **accurate** (not grossly conservative over estimates) site requirements for the MCR system (including the visualization system) at proposal submission time. The University will be responsible for supplying the external elements of the power, cooling, and cable management systems.

The systems will be physically located inside an Property Protection Area. Dialup capability and internet access to system will be allowed. Authorized individuals may be allowed remote access for running diagnostics and problem resolution. Interaction of the on-site engineering staff with factory support personnel may be limited in some ways (e.g., dissemination of memory dumps from the system may be restricted). These limitations emphasize the importance of local access to source code, particularly for operating system daemons. All on-site personnel will require being DOE P-cleared or P-clearable. It will be difficult to provide access to foreign nationals.

On-site space will be provided for personnel and equipment storage.

A safety plan will be required for on-site personnel. They will be expected to practice safe work habits, especially in the areas of electrical, mechanical, and laser activities.

End of Section 1

2 MCR Strategy and Architecture

This section describes the overall MCR hardware and software strategy and architecture, Linux development and support strategy and outlines a plan for MCR build.

2.1 LC Hardware and Software Strategy and MCR Architecture

The University's scalable systems strategy (known as the Livermore Model, Figure 2.1-1) is to have a unified software environment available on all cluster systems we put into production. The main purpose of this strategy is to enable highly complex scientific simulation applications to be portable across multiple platforms at any given point in time and to provide a stable target environment over multiple generations of platforms. This strategy has been successful in providing a stable target applications environment since about 1992, when the Meiko CS-2 MPP was introduced at LLNL.

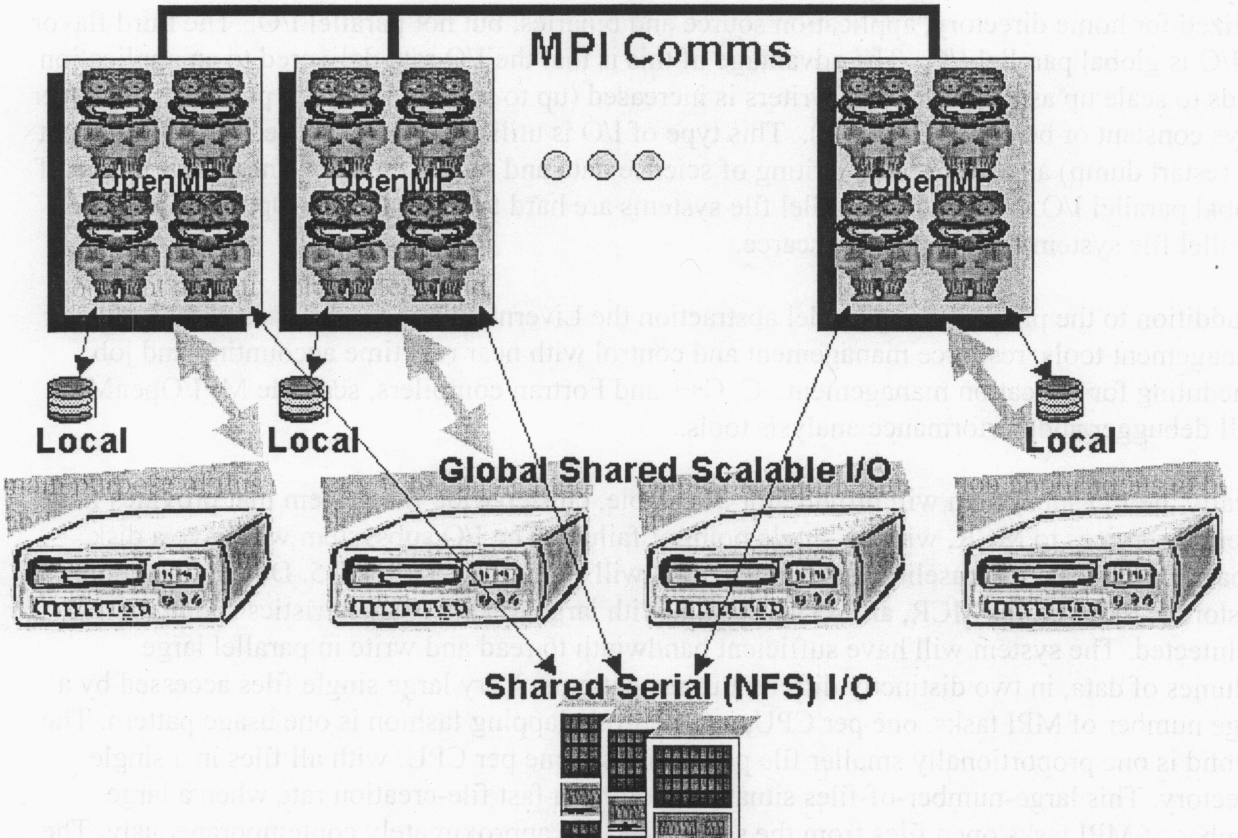


Figure 2.1-1: The Livermore Model provides a stable target environment for scientific simulation codes by abstracting the parallelism and I/O model. Parallelism is MPI tasks exchanging data over a high-speed, low latency communication mechanism and a small number of OpenMP threads per MPI task. This model includes an OS on every node and three types of I/O. It also includes C, C++ and Fortran compilers, TotalView debugger and node hardware and MPI/OpenMP performance analysis tools.

The basic idea of the Livermore model is to abstract the parallelism and I/O model of the computing environment. At a high enough level of standards based abstraction, the computing environment evolves fairly slowly and most machines of a given era are roughly equivalent. The parallelism abstraction is based on shared memory multiprocessors (SMPs) interconnected with a high-speed, low-latency interconnect. Each SMP has a hierarchical shared memory: processor registers; on-chip and off-chip caches; (possibly NUMA) memory. Applications must thus utilize MPI to communicate between the distributed memory elements. In addition, each MPI task can utilize compiler generated OpenMP threads for on SMP parallelism. Each SMP or node is assumed to have a local, full functioning POSIX compliant operating system. In addition, to the local disk for OS swapping, applications use highly scalable I/O by writing to the local disk. The drawback of this flavor of I/O is that it is local: data must be migrated to the local disk before application execution and retrieved after execution. Thus, local I/O is predominantly used for intermediate, temporary results. A second flavor of I/O that is heavily utilized is global serial (NFS) I/O. This has the advantage of being global, but the disadvantage that the performance only scales to the limit of a single NFS server (currently 20-100 MB/s). This type of I/O can be utilized for home directory, application source and binaries, but not parallel I/O. The third flavor of I/O is global parallel I/O. The advantage of this is that the I/O rate delivered to an application tends to scale up as the number of writers is increased (up to a point, and then performance either stays constant or begins to decrease). This type of I/O is utilized for parallel reading of the input (or restart dump) and for parallel writing of science data and restart dumps. The disadvantage of global parallel I/O is that good parallel file systems are hard to come by and Open Source parallel file systems are even more scarce.

In addition to the programming model abstraction the Livermore Model assumes parallel cluster management tools, resource management and control with near real time accounting and job scheduling for allocation management. C, C++ and Fortran compilers, scalable MPI/OpenMP GUI debugger and performance analysis tools..

Ideally the I/O subsystem will provide for a scalable, cluster-wide file system that provides fault-tolerant services to MCR, with no single point of failure. The I/O subsystem will have a disk capacity of at least the baseline requirement, and will support RAID level 5. Due to the amount of storage required for MCR, an I/O subsystem with large MTBF characteristics should be architected. The system will have sufficient bandwidth to read and write in parallel large volumes of data, in two distinctly different usage patterns. Very large single files accessed by a large number of MPI tasks, one per CPU, in a non-overlapping fashion is one usage pattern. The second is one proportionally smaller file per MPI task, one per CPU, with all files in a single directory. This large-number-of-files situation requires a fast file-creation rate when a large number of MPI tasks open files from the same directory approximately contemporaneously. The I/O subsystem will also support a scalable parallel file system accessible from every node in the system. The ideal I/O subsystem will also be capable of providing services to requests beyond the MCR cluster, indicated in Figure 2.1-2 by the yellow switch bar that extends outside the cluster-wide box. As the parallel file system is a critical system resource, it will be highly reliable. For example, the Lustre Object File System is expected to provide the desired file-system characteristics, and will leverage ASCI PathForward investments in open-source software.

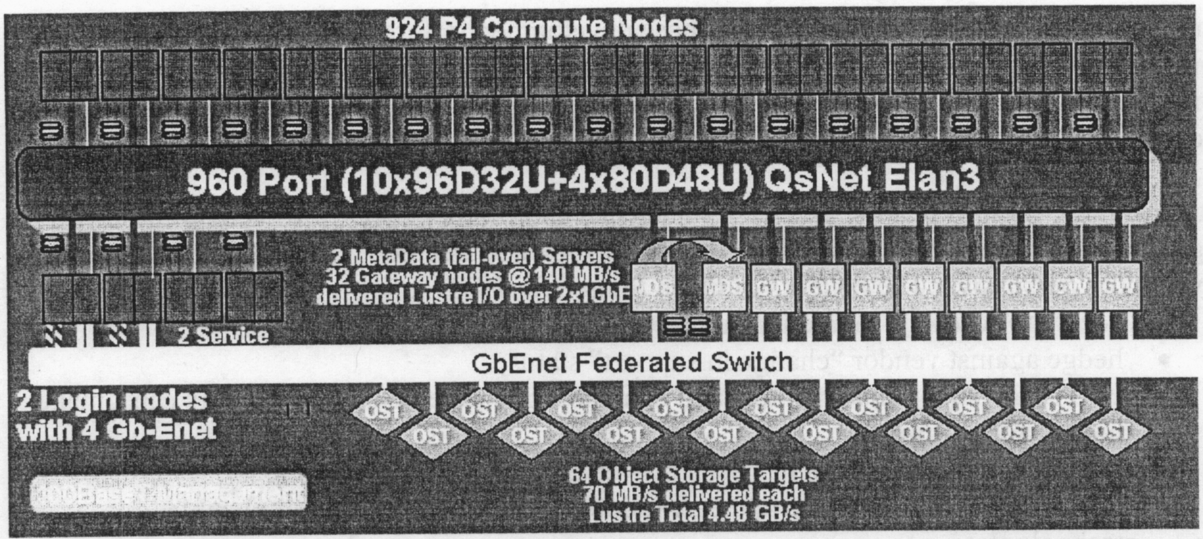

924 P4 Compute Nodes
960 Port (10x96D32U+4x80D48U) QsNet Elan3
2 MetaData (fail-over) Servers
32 Gateway nodes @ 140 MB/s
delivered Lustre I/O over 2x1GbE
MDS
MDS
GW
GW
GW
GW
GW
GW
GW
GW
2 Service
GbEnet Federated Switch
2 Login nodes
with 4 Gb-Enet
OST
64 Object Storage Targets
70 MB/s delivered each
Lustre Total 4.48 GB/s

Figure 2.1-2 MCR SAN Architecture. MCR system architecture includes clustered I/O model, local node disks, dedicated login nodes, dedicated visualization nodes and compute nodes and network attached RAID disk resources. Scalable user applications (either batch or interactive) will only run in the "parallel batch/interactive nodes" compute partition. The login nodes host interactive login sessions and code development activities, but not for running MPI jobs. The visualization nodes host parallel batch and interactive visualization jobs.

It is our intention to use MCR system as the vehicle for providing the first Storage Area Network (SAN) in to the LLNL unclassified computing infrastructure, see Figure 2.1-2. Therefore, it is essential that the I/O subsystem for connections for MCR be based on 1000Base-SW. Our strategy is to accrete other systems (e.g., IBM SP based High Performance Storage System) to this SAN environment after MCR is integrated. The aim here is to have MCR provide scalable cluster wide parallel file system service for all OCF resources over time. Thus the cluster wide file system (CWFS) supplied with MCR should be able to provide scalable global parallel performance to the other resources mentioned.

2.1.1 LC Linux Strategy for HPTC Scalable Clusters

Our strategy is to extend the Livermore model from proprietary systems in use at LLNL (e.g., IBM SP with AIX and PSSP, Compaq Sierra with TruUNIX) to commodity (i.e., IA-32) nodes with Linux. The first implementation of this strategy was the Parallel Capacity Resource (PCR) procurement last year. That procurement produced two clusters (one 128 dual P4 node and one 88 dual P4 node) that are now migrating into production usage in the Secure Computing Facility (SCF) for use by the ASCI on-going computing element.

Over the past five years, the Open Source community development model, popularized by the GNU project and the Linux OS development effort, has shown remarkable capability to deliver freely available software that satisfies a broad range of computing requirements in an astounding range of computing environments: from desktops to high-availability configurations and embedded systems to teraFLOP/s clusters. These efforts have had a significant impact on the high performance technical computing landscape as witnessed by the fact that all major computer system manufacturers now offer Linux solutions. In the HPTC (high performance technical computing) environment, the Open Source movement has created an environment where multiple organizations can contribute software development and enhancements to cluster solutions. These development efforts have reached a critical mass and are now producing multiple cluster offerings that are competitive with other vendor proprietary solutions. In addition, the price performance of these solutions, when based on commodity hardware components, is extremely attractive for HPTC sites.

It is for these reasons that the University launched exploratory Linux efforts over three years ago. During this period of time, the power of the Open Source development model became even more persuasive. The benefits of Open Source for HPTC sites include:

- access to software source for quick bug fix ability;
- hedge against vendor "change in support" status;
- HPTC site tend to have unique and demanding requirements that vendors can't make money supporting;
- joint development model has been proven effective multiple times;
- HPTC sites are developers and early adopters of critical, widely applicable technologies;
- multiple Open Source HPTC sites find more bugs;
- the perception that Open Source has become the next wave of software development by increasing return on investment for companies that figure out how to leverage "free software";
- fosters community development model that leads to accelerated innovation through competition.

Livermore Computing is actively pursuing an Open Source development model to leverage these astounding benefits in a "Generally Available" or Production computing resource. To this end, our strategy is to focus on high performance, scalable cluster computing environments with as much Open Source software as possible. The deployment strategy for Open Source technology is to start with small clusters and grow the number and size of clusters based on Open Source technology in Production over time. Livermore Computing has demonstrated that this is a viable approach through the successful development of the Parallel Capacity Resources (PCR), a Linux-based set of clusters first deployed at the end of FY2001. With PCR, we used this strategy to fill the "capacity computing for capability (MPI parallelized) jobs" niche. The first step of this strategy allowed LC the flexibility, while still meeting SSP programmatic objectives, to start with small sized clusters and then grow the capacity environment by adding clusters and by increasing the size of the clusters deployed over time as the Open Source software technology matures and LC gets more proficient at deploying Linux clusters into Production. This procurement represents the next step in this direction with the development of a capacity resource based on an open-source development model..

The successful migration of LC computing to the commodity hardware and Open Source software technology can only be completed with LC fully contributing to and fully engaging in the community development and support model. However, this is only one aspect of the overall strategy. Past large-system deployments at LC have all been accomplished via long-term relationships with computing system manufacturers in the form of partnerships. This is a key methodology that LC has utilized to build the University's existing unclassified Multi-programmatic and Institutional Computing (M&IC) environment.

The current MCR effort builds on that vendor partnership model and represents a solicitation for partnering between LC and a vendor partner in Open Source development efforts described below or those of interest to the partner. From market survey discussions in preparation for this RFP, it became clear that the cluster strategy espoused by LC and concretized by this procurement represents a "productizable" model that multiple potential vendor partners find highly attractive. Thus, these Open Source cluster efforts with a vendor partner should accomplish the following long-range goals:

- enhance the state-of-the-art in high-end clusters
- provide competitive products for the vendor partner
- provide cost effective Production clusters to accomplish LLNL M&IC and NNSA SSP program goals

There remain challenges in the HPTC Linux cluster environment. Most prominent challenge is, including the lack of an Open Source, scalable, high-performance parallel file system. The second most prominent challenge is the lack of effective cluster scheduling (including checkpoint restart and GANG scheduling) technology. We intend to address these issues with future Open Source development in this partnership.

2.1.2 MCR Hardware Architecture

The above M&IC requirements for an extremely cost-effective large scale scientific computing platform lead Livermore Computing to large cluster architecture based on IA-32, Quadrics QsNet and BlueArc OST. IA-32 based nodes were selected after running a series of M&IC applications benchmarks that indicate that Pentium 4/Xeon Foster and Prestonia processors deliver the best performance and cost performance of any option available. As indicated in section 1, M&IC applications are floating point arithmetic and memory bandwidth intensive. However, given that IA-32 bus bandwidth at 3.2 GB/s (1.7-1.9 GB/s delivered) in Foster/i860/RDRAM or Prestonia/(e7500|ServerWorks GrandChampion LE)/PC200 DDRSDRAM based motherboards is exhausted for most M&IC applications with a one or two active processes or HyperThreads (a few can utilize more processors), dual nodes have been selected by M&IC users for the MCR. Therefore, Offerors are encouraged to bid dual nodes.

M&IC applications are quite demanding on delivered interconnect latency and bandwidth. Thus, Quadrics QsNet Elan3 was selected because it delivers high bandwidth (>300 MB/s) and low latency (<5.0 μs) at commodity interconnect pricing. In addition, M&IC plans to run the MCR as a combined capability and capacity machine and therefore a reduced minimum bisection bandwidth can be tolerated. Thus we have chosen a QsNet configuration with 128-port (128-way) switch elements constructed as a two stage, fat-tree, federated switch configuration with ten first level and four second level switches: a total of fourteen 128-way switches (see Figure 2.1-3). The first level switches are configured with 96 ports for nodes and 32 ports for connecting to other QsNet switches (96D32U). The second level switches are configured with 96 ports for first level switches and 48 unused ports (96D48U). This configuration costs less than a 64D64U level one switch configuration, at the expense of reduced bisection bandwidth.

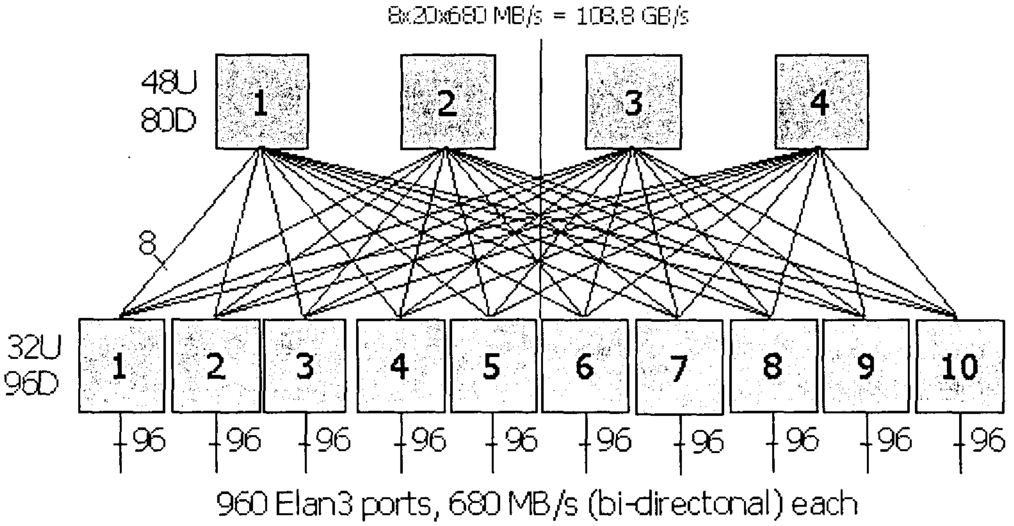

8x20x680 MB/s = 108.8 GB/s
48U
80D
1
2
3
4
8
32U
96D
1
2
3
4
5
6
7
8
9
10
96
96
96
96
96
96
96
96
96
96
960 Elan3 ports, 680 MB/s (bi-directonal) each

Figure 2.1-3 Quadrics QsNet Elan3 for MCR is based on 14 128-port Elan3 switches configured as a two stage fat-tree federated switch. Each of the 10 Elan3 first stage switches is configured with 96 ports for nodes and 32 for the second level switches (96D32U). The 4 second level switches are configured with 96 ports for connecting the first level switches and 48 unused ports (96D48U).

This switch configuration leads to a natural scalable unit size of ninety-six nodes. Thus, the University envisions building the MCR in ten equal size scalable units of ninety-six nodes each. From Figure 2.1-2 it is clear that a vast majority of the MCR nodes are compute nodes. Hence if we define a single scalable unit, to be built first, called the First Scalable Unit (FSU) with the gateway nodes, Login nodes, management nodes (not on QsNet switch) and MDS (fail-over pair) nodes and enough compute nodes to fill out the remainder of the first scalable unit. Then the nine remainder scalable units are identical. These nine scalable units are called Compute Node Scalable Unites (CNSU)

The Lustre Lite file system, described in the software section 2.2.5 below, has several ramifications on the MCR hardware architecture. First, Lustre Lite has file system clients that provide global file system access on every compute node. This implies that the high speed, low latency communication mechanism for the file system is QsNet. Second, Lustre Lite requires a Meta Data Server (MDS) fail-over pair. These MDS nodes must be configured to support Kimber Lite Linux High Availability fail-over schemes. This means a dedicated TTY and 100BaseT Ethernet for heartbeat and shared disk for metadata. The MDS needs about 2.5% of object storage target capacity (or about 2.5 TB of usable RAID5 disk) for the meta data accessible at 256 MB/s delivered of 512B block raw read/write (or 500x512B I/O's per second) performance from either node. 2.0Gb/s FibreChannel-attached RAID5 disks are ideal for this. Third, as Lustre Lite migrates to full Lustre, the Lustre file system will be extended beyond the MCR cluster boundary. An extensible, interoperable, commodity SAN technology is required for extending Lustre into this heterogeneous environment. We plan to integrate the Federated Gb Ethernet switch infrastructure in FY03. Until then, the Lustre Lite OST will be direct attach or small port count Gb Ethernet switch connected for Gateway node fail-over tolerance. Fourth, Lustre Lite requires Object Storage Targets (OST) to manage creating, locking, writing, reading and deleting of objects (aggregations of disk blocks). The chosen SAN-based OST (NAS) for MCR is the BlueArc Silicon Server Si7500 with OST enhancements on 1000Base-SW networking.

2.2 LC Software Environment for Linux Clusters

To execute the Livermore Computing Linux software strategy, LC provides a complete software environment for Linux clusters called CHAOS that meets the programmatic and operational requirements described in the sections above. In addition, Livermore Computing currently is actively involved in the Open Source development of Linux clustering tools, the SLURM resource manager, the DPCS metabatch scheduler and resource accounting system, and the Lustre cluster wide file system. These components and the rest of the CHAOS environment will be installed on the MCR cluster after it is delivered to LLNL (see section 6.3 for milestone details).

2.2.1 Clustered High Availability Operating System (CHAOS)

Livermore Computing produces and supports CHAOS (Clustered High Availability Operating System, http://www.llnl.gov/linux/chaos), a cluster operating environment based on RedHat Linux. At the core of CHAOS is a RedHat "boxed set" distribution. Some components of that distribution are modified to meet the demands of high performance computing and the Livermore Computing center. A number of additional cluster-aware components are added on.

A CHAOS distribution contains a set of RPM (RedHat Package Manager) files, RPM lists for each type of node (compute, management, Gateway, and login), and a methodology for installing and administering clusters. It is produced internally and therefore supports a short list of hardware and software. This focus on the Livermore Computing environment permits the laboratory to support CHAOS with a small crew and to be agile in planning its content and direction.

In addition to the products of Open Source development described below and the RedHat boxed set distribution, CHAOS includes the following software components:

- *kernel* - The CHAOS kernel is based on a RedHat kernel with additions in the areas of VM/device support for Quadrics Elan3, ECC and FLASH memory device support for i860 chipset, MCL crash dump support, miscellaneous bug fixes, and optimized configurations for our hardware.
- *Quadrics RMS, libelan, MPI, etc.* - The Quadrics software environment used to run parallel programs.
- *Crash* - The Mission Critical Linux crash dump analysis suite is used to examine post mortem contents of a kernel crash dump.
- *lm_sensors* - Hardware monitoring, linked to the SNMP host monitoring system, monitors motherboard chipset sensors such as temperature, fan speed, and power supply voltages.
- *fping* - Fping is a rudimentary node status tool which can ping nodes in parallel. In combination with genders tools, fping can quickly find nodes in the cluster that are turned off or otherwise unreachable on management network.
- *OpenSSH* - OpenSSH provides encrypted remote login/shell service that is PAM-aware.

- *PAM Tools* - PAM modules for One Time Passwords (OTP), Kerberos V, and RMS (to authorize a user's access to a compute node only when RMS is running that user's job) are used to leverage Livermore Computing's DCE and OTP infrastructure for PAM-aware applications.
- *Firmware* - Firmware images for motherboards, including FLASH/CMOS support software is included in CHAOS. Firmware and support software for power control/serial console hardware is also included.
- *MPI Test Suite* - The Pallas MPI Benchmark (PMB), Effective Bandwidth test (BEFF), and Quadrics MPI ping-pong test (mping) are packaged with CHAOS with a script to maintain a continuous MPI workload under RMS for testing purposes.
- *NTTCP* - The NTTCP TCP bandwidth test is packaged along with genders-aware scripts that can simulate load on the management ethernet for testing purposes.
- *super* - Super extends administrative privileges to non-root users.
- *Intel Compilers* - The Intel IA32 FORTRAN, C and C++ compilers.
- *PGI Compilers* - The Portland Group Fortran and C Compilers
- *Totalview* - The TotalView parallel Debugger from Aetnus.
- *Other Libraries/Tools* - Other libraries and tools such as Atlas, COG, NDF, netCDF, Hyper, ScaLAPAC, OpenGL, Yorick, Silo, VTK, and Findentry are maintained on Livermore Computing Linux systems.

2.2.1.1 CHAOS Status

CHAOS version 0.1 currently runs on the PCR (Parallel Capacity Resource) systems procured in Q3CY2001. These are 26-, 88-, and 128- way clusters based on dual 1.7Ghz Pentium 4/Xeon nodes and Quadrics Elan3 interconnect. Version 0.1 is based on RedHat 7.1 and the RedHat 2.4.9 errata kernel series.

CHAOS 1.0, the first “official” CHAOS release to be subject to formal integration testing, will run on the PCR clusters in the May-June 2002 time frame. It will be based on RedHat 7.3 and the RedHat 2.4.18 kernel.

CHAOS will be extended to operate the MCR cluster as needed depending on Offeror’s proposal (e.g., power and console management solutions).

2.2.2 LLNL Cluster Tools

The following Open Source cluster tools (http://www.llnl.gov/linux/ctp) are under active development and have been deployed on the PCR clusters:

- *pdsh* - The Parallel Distributed Shell utility executes processes across groups of nodes in parallel. It is also capable of running small MPI jobs on the Elan3 interconnect.
- *YACI* – Yet Another Cluster Installer is Livermore's system installation tool based on various cluster installers such as VA system imager and LUI that can fully install an 88-node cluster in about 10 minutes. It is image-based and uses an NFS pull from many network-booted nodes running in parallel.
- *Genders* - is a static system configuration database and rdist Distfile preprocessor. Each node has a list of "attributes" that in combination describe the configuration of the node. The genders system enables identical scripts to perform different functions depending on their context. An rdist Distfile preprocessor expands attribute macros into node lists.

- *ConMan* - The ConMan console manager manages serial consoles connected either to hardwired serial ports or remote terminal servers (telnet based). Performs logging of console output, and manages interactive sessions, permitting console sharing, console stealing, console broadcast, and interfaces for transmitting a serial break or resetting a node via PowerMan.
- *PowerMan* – The PowerMan power manager manages system power controllers and is capable of sequenced power on/off for groups of nodes and initiating reset (both plug off/on and hardware reset if available). Powerman currently supports the Linux Networx ICE box, WTI RPC's, and the API Networks modified Wake-on-lan. It can be extended to support new hardware.
- *Host Monitoring System* – Livermore Computing's SNMP based host monitoring system stores current state in a MySQL database and long term state in an RRD (round robin database). Collection software polls cluster nodes in parallel using SNMP bulk queries and a sliding window algorithm to reduce polling latency. Status is presented via web using Apache and PHP.

2.2.3 Simple Linux Utility for Resource Management

Simple Linux Utility for Resource Management (SLURM) is an Open Source, fault-tolerant and highly scalable cluster management and job scheduling system for clusters containing thousands of nodes. SLURM is presently under design and development at LLNL. (`http://www.llnl.gov/liv_comp/SLURM/SLURM_home.html`).

The primary functions of SLURM are:

- Monitoring the state of nodes in the cluster
- Logically organizing the nodes into partitions with flexible parameters
- Accepting job requests. While SLURM can support a simple queuing algorithm. DPCS will manage the order of job initiations through its sophisticated algorithms described in section 2.2.3 of this document.
- Allocating both node and interconnect resources to jobs.
- Monitoring the state of running jobs including resource utilization rates.

SLURM will utilize Kerberos V5 based authentication. The design also includes a scalable, general-purpose communications infrastructure. APIs will be available to support all functions for ease of integration with external schedulers. SLURM is written in the C language, with a GNU autoconf configuration engine. While initially written for Linux and Quadrics Elan3 interconnects, our design calls for ease of portability. We anticipate having a functional version of SLURM available in August 2002.

2.2.4 Distributed Production Control System (DPCS)

The Distributed Production Control System (DPCS) is an Open Source product of the Livermore Computing (LC) Center. (`http://www.llnl.gov/liv_comp/DPCS/DPCS_home.html`) The DPCS project began in 1991 when LC started to convert all of its production computer systems to UNIX platforms. DPCS was in production in October 1992 and has continued to develop since then.

The primary purpose of DPCS is to allocate computer resources, according to resource delivery goals, for LC's UNIX-based production computer systems. This is accomplished through a complex hierarchy of:

- computer-share bank accounts
- time-usage monitoring tools
- run-control mechanisms

DPCS lets organizations control who uses their computing resources and how rapidly those resources are used. It also manages an underlying batch system that actually runs production jobs guided by DPCS policies.

Resource Delivery Goals: Defined by LC management in coordination with program managers. Program managers oversee a group's access to production computer system resources. These goals are "programmed" into the DPCS system which then attempts to meet those goals. In other words, DPCS is used to assure that the right people, projects and organizations get appropriate access to a center's computer resources. The DPCS contains two major subsystems that work together to deliver resources as required. The Resource Allocation & Control System (RAC) provides mechanisms for allocating machine resources among diverse users and groups, while the Production Workload Scheduler (PWS) provides mechanisms for automatically scheduling batch (production) jobs on the machines.

The RAC system is used to declare production hosts, to create and manage recharge accounts, resource allocation partitions, resource allocation pools, and user resource allocations within the resource allocation pools. Caveat Emptor: a recharge account should not be confused with a user login account. DPCS uses the term bank as a synonym for "resource allocation pool."

The Production Workload Scheduler is a set of daemons and utilities that work with the RAC system to schedule batch jobs on the DPCS production hosts. It employs policy as instructed through the mechanisms provided to DPCS managers and resource managers to prioritize and schedule production appropriately.

2.2.5 Lustre Lite Cluster Wide File System

The University plans to utilize the Lustre Lite Cluster Wide File System on the MCR cluster. To that end, Livermore Computing and Cluster File Systems, Inc. have been actively engaged in developing a "lite" version of Lustre to run on MCR this summer. See http://www.lustre.org/ for more information on Lustre.

Lustre Lite's impact on the MCR architecture is substantial. This solicitation includes two meta data server (MDS) nodes in fail over configuration and about 2.5 TB of disk for Lustre Lite meta data. In addition, the University will supply 64 Object Storage Targets (OST) with a combined I/O rate of 4.48 GB/s, 100 TB of RAID5 disk attached to the cluster via GbEthernet. The specified configuration includes thirty two gate way nodes with two GbEthernet adapters and one QsNet adapter to gateway Lustre file system data (both meta data and objects) between the OST, MDS and Lustre Clients (compute and login nodes).

Since Lustre Lite is under active development one of the first activities envisioned for the MCR cluster during factory build is the scaling testing and performance tuning of Lustre Lite. To this end the First Scalable Unit (FSU) should be built first (hence the name) and attached to the GFE OST infrastructure to facilitate this testing activity as early as possible.

2.2.6 The Livermore Computing Linux Cluster Support Model

Livermore Computing Open Source developers (Cluster Tools, DPCS, SLURM, Lustre Lite) work closely with system administrators and users to resolve problems on production systems. For any given software package, there is a designated package owner who is charged with handling any support issues that arise in production. Depending on the nature of the package, owners may be the primary developer and fix bugs themselves, or they may be the liaison to an external support resource.

External support relationships are primarily developer-to-developer. In the case of RedHat, we have a full-time RedHat engineer on site who can work directly with Livermore systems and support people, and act as the liaison to RedHat for everything in the RedHat boxed set. In the case of Quadrics, an on-going cooperative research and development agreement (CRADA) and a support contract are leveraged to get bugs fixed in production.

2.2.7 Integration Testing

Each CHAOS release is subject to integration testing that includes regression tests for past problems, basic functionality tests, and real users applications. Each of the software components is developed asynchronously, but come together in system (CHAOS) releases and separate package (DPCS, Lustre Lite, SLURM, Cluster Tools) releases. Due to this separation, system and package testing and installation on production clusters can be scheduled and executed independently. A 26-node development cluster called DEV which can be rapidly reinstalled into any past CHAOS environment as well as new prototype environments, is available for unit testing of individual software components, integration testing of a complete CHAOS release, and debugging of defects that arise in production. This cluster, along with the project CVS repository, can accommodate external collaborators working with Livermore Computing on the Open Source projects described above.

2.3 MCR Build Strategy

The MCR build strategy is based on the scalable unit concept defined in section 3.2.7. The Offeror will build the First Scalable Unit (FSU, see section 3.2.7.2) and install a software image mutually agreed between the Offeror, Quadrics, and Cluster Filesystems, Inc. and approved by the University technical representative. The FSU, which contains all the login, management, gateway, Lustre Meta Data servers and disk, and a complement of compute nodes, will undergo initial functionality and performance testing and then be used as the vehicle to scale up Lustre Lite as MCR is built.

After the FSU is complete, the Offeror will assemble the remainder of the cluster consisting of nine Compute Node Scalable Units (CNSU, see section 3.2.7.3) by adding one or more CNSU to the Quadrics Federated QsNet switch and allowing time for Lustre Lite scaling testing.

For the cases where the Offeror is being asked to integrate Government Furnished Equipment (GFE), said equipment will be provided to the Offeror prior to commencing the FSU build.

Specifically, the Quadrics QsNet and 1000Base-SW BlueArc OST's will be provided for the MCR build in mid June 2002. 16 BlueArcs will be provided in mid June with the reaming 48 shipped to Livermore for assembly after MCR delivery.

Once the MCR is built and Lustre Lite scaling to 960 nodes is complete, pre-ship testing commences. This level of testing consists of running a specific set of parallel applications across the machine. Once the exit criteria defined in section **Error! Reference source not found.** are met, the machine is disassembled and shipped to the LLNL site and reassembled. At LLNL, the post-ship test is re-run to verify that the hardware survived disassembly, shipment and reassembly.

Once the hardware has been reassembled and passed the post-ship test and turned over to University personnel, the University will install the CHAOS environment with the aid of the Offeror personnel. Acceptance testing will take place in the CHAOS environment as described in section 6.3.9.

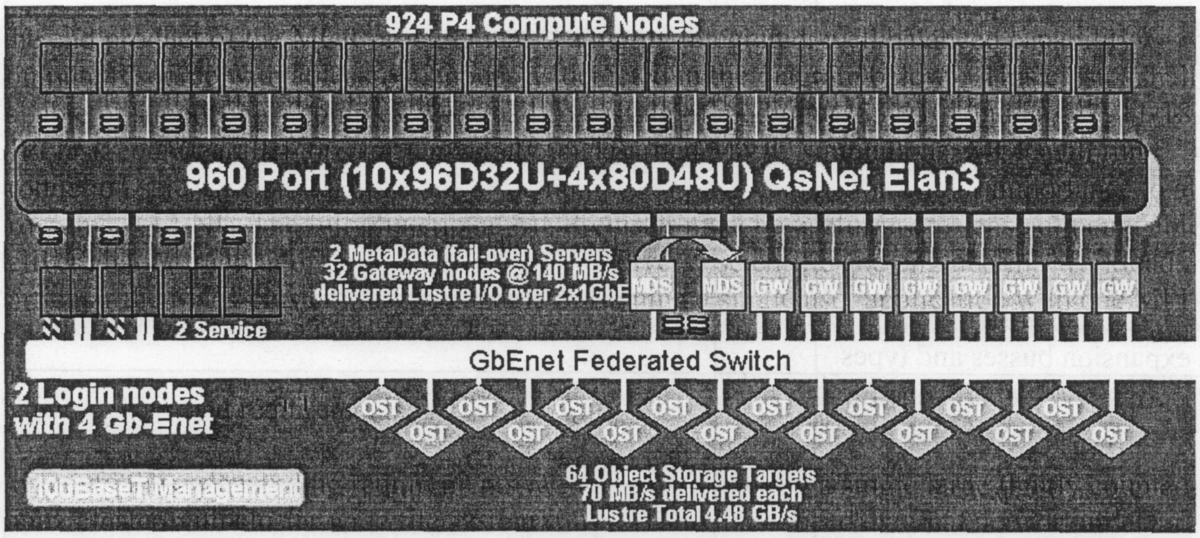
The CHAOS environment may require modification to run on the target hardware. This development work will take place in parallel with the initial FSU build using "early ship" nodes and serial/power management hardware delivered to Livermore as described in section **Error! Reference source not found.**.

End of Section 2

3 MCR Technical Requirements

The end product of the MCR procurement is a highly integrated, well balanced capability compute resource with at least 962 nodes as depicted in the diagram below. This cluster shall be capable of supporting a complex workload consisting of medium (455-910) and large (911-1820) MPI task count parallel jobs for University unclassified M&IC simulations. MCR will run production scientific simulations of a wide number of physical phenomena of importance to all programs at LLNL. The fully functional MCR cluster must be useful in the sense of being able to deliver a large fraction of peak performance to a diverse scientific and engineering workload. In particular the MCR cluster must be capable of running a single user application with one MPI task per CPU over all 924 compute nodes in the system. The MCR cluster must also be useful in the sense that the code development and production environments are robust and facilitate the dynamic workload requirements.

To satisfy these demanding requirements, we anticipate needing a single large tightly coupled Linux cluster with cluster wide file system (CWFS) and high-speed external networking. Our requirement is to have this cluster built from commodity nodes containing two Intel Pentium 4 Xeon microprocessors.. The University will furnish QsNet ELAN3 switches, cables and adapters for the MCR clusters. Additionally, the University will furnish the OST and GbEnet Federated Switch required to support the CWFS.



The specific hardware and software Mandatory Requirements the MCR cluster shall meet are delineated with (MR) designation. Mandatory options (MO) are requirements whose availability as an option the University deems necessary. Hardware and software technical Target Requirements that are desirable, but not mandatory, are delineated and prioritized with (TR-k, k=1,2 or 3) designation.

In addition to the mandatory hardware and software requirements, the Offeror will deliver any Target Requirements for the MCR clusters, and any additional features consistent with the objectives of this project and Offeror's project plan, which the Offeror believes will be of benefit to the partnership.

3.1 High-Level Hardware Summary (TR-1)

This section will contain a detailed description of the proposed MCR cluster. The features and functionality of all major components of the system shall be discussed in detail. The Offeror will provide an architectural diagram of the MCR cluster, labeling all component elements and providing bandwidth and latency characteristics (speeds and feeds) of and between elements. The Offeror will provide an architectural diagram for each MCR node type bid, labeling all component elements and providing bandwidth and latency characteristics (speeds and feeds) of and between elements. The node architectural diagrams will specifically show and label the chip set used and denote independent PCI buses and slots and label these with bus widths and speeds.

MCR Cluster Requirements Summary Matrix

The following matrix identifies the highest priority technical requirements (TR-1) and will be completed in its entirety. Entries label by the University as Does Not Apply (DNA), need not be filled in. All entries shall be labeled N/A if the requirement is not offered. In addition, the system requirements summary matrix will be completed for any alternate proposed systems submitted.

Index	Requirement Description	Qty	Offeror response
1	Compute node product designation		
2	Compute node processor type, speed and L2 cache size		
3	Compute node memory bus type and speed		
4	Compute node chip set designation		
5	Compute node number of expansion busses and types		
6	Compute node number and type of expansion bus slots for each bus		
7	Type and size of compute node memory		
8	Type and size of compute node local disk		
9	Login node product designation		
1	Login node processor type, speed and L2 cache size		
1	Login node memory bus type and speed		
1	Login node chip set designation		
1	Login node number of expansion busses and types		

Index	Requirement Description	Qty	Offeror response
1	Login node number and type of expansion bus slots for each bus		
1	Type and size of login node memory		
1	Type and size of login node local disk		
1	GW node product designation		
1	GW node processor type, speed and L2 cache size		
1	GW node memory bus type and speed		
2	GW node chip set designation		
2	GW node number of expansion busses and types		
2	GW node number and type of expansion bus slots for each bus		
2	Type and size of GW node memory		
2	Type and size of GW node local disk		
2	MGMT node product designation		
2	MGMT node processor type, speed and L2 cache size		
2	MGMT node memory bus type and speed		
2	MGMT node chip set designation		
2	MGMT node number of expansion busses and types		
3	MGMT node number and type of expansion bus slots for each bus		
3	Type and size of MGMT node memory		
3	Type and size of MTMT node local disk		
3	MDS node product designation		

Ind ex	Requirement Description	Qty	Offeror response
3	MDS node processor type, speed and L2 cache size		
3	MDS node memory bus type and speed		
3	MDS node chip set designation		
3	MDS node number of expansion busses and types		
3	MDS node number and type of expansion bus slots for each bus		
3	Type and size of MDS node memory		
4	Type and size of MDS node local disk		

3.2 MCR Hardware Requirements

3.2.1 MCR Scalable SMP Cluster (MR)

The Offeror shall provide a cluster of at least 962 Pentium 4 Xeon dual processor nodes. There will be five node types. at least 924 compute nodes; 32 gateway nodes; 2 login nodes, 2 Lustre Meta Data Servers (MDS) and 2 management nodes. All nodes with the exception of the management nodes shall be attached to the QsNet ELAN3 network. All nodes shall be attached to the management Ethernet network. Additionally, the gateway nodes will attach to the GFE Lustre OST devices via GFE GbEthernet.

3.2.2 MCR Node Requirements

The following requirements apply to all node types except where superceded in subsequent sections.

3.2.2.1 Processor and Memory Interface (TR-1)

The MCR nodes will be configured with dual Pentium 4 Xeon with at least 256KB of L2 2.2 GHz or faster processors.

3.2.2.2 Later Generation Processor (TR-2)

If available, 2.4GHz processors with 512KB of L2 cache with Jackson (or HyperThreading) technology will be supplied.

3.2.2.3 Chip Set and Memory Interface (TR-1)

The MCR nodes will be configured with chip sets compatible with PC600/800 RDRAM or PC-200 DDR SDRAM and PCI 66/64 or PCI-X. It is highly preferred that the chip set for all nodes in the cluster be the same.

3.2.2.4 Node Delivered Memory Performance (TR-1)

The MCR nodes will be configured to deliver at least 1.8 GB/s aggregate memory bandwidth when running one copy of the streams benchmark per CPU or HyperThread, if applicable. Offeror will report with proposal the delivered streams performance running one copy of the streams benchmark per CPU or HyperThread, if applicable, for each bid MCR node type. See www.llnl.gov/asci/purple/benchmarks/ for the streams benchmark.

3.2.2.5 Node Delivered PCI Performance (TR-1)

The MCR compute nodes and chip set will be configured to deliver at least 300 MB/s PCI 64b/66 MHz bandwidth when running the MPI bandwidth benchmark across the Quadrics QsNet ELAN3 PCI adapter. Offeror will report with proposal the compute node delivered streams and simultaneous MPI bandwidth benchmark over ELAN3 PCI 64b/66 MHz adapter performance. For maximum QsNet performance, the ServerWorks GrandChampion HE or Intel i860 or e7500 chipset is preferred, but not mandatory.

We recommend the "com" benchmark from the ASCI Purple Presta MPI Stress Test suite. Presta can be obtained from:
http://www.llnl.gov/asci/purple/benchmarks/limited/presta/

3.2.2.6 Node Memory Size (TR-1)

The MCR nodes will be configured with at least 2.0 GiB of memory (1.0 GiB of memory per processor).

3.2.2.7 Node Memory Error Protection and Detection (TR-1)

The MCR nodes will be configured with single bit correction and double bit detection (SECDED) error checking and correcting (ECC) registered memory or better. The hardware will detect and count single and double bit memory errors on each memory RIMM or DIMM (lowest level memory FRU). The operating system memory error kernel module in section 3.3.2 and diagnostics in section 4.5.2 will interface to this hardware memory error detection facility.

3.2.2.8 Local Disk (TR-1)

The MCR cluster nodes will have a single hard disk drive (HDD). The local disk interface can either be ATA100 EIDE or UltraSCSI160 or equivalent or faster. Front panel access to the local disk is desirable. If the bid local disk is ATA100 or equivalent or faster, then the local disk will have a capacity of 120 GB or more and rotate at 7,200 RPM or faster. If the bid local disk is UltraSCSI160 or equivalent or faster, then the local disk will have a capacity of 72 GB or more and rotate at 10,000 RPM or faster and will be hot swappable.

3.2.2.9 Node Form Factor (TR-1)

The Offeror will provide nodes packaged in standard 19" rack mountable enclosures with at most 2U form factor per node. It is highly desirable that the compute nodes be at most 1U form factor.

3.2.2.10 Integrated Management Ethernet (TR-1)

The MCR cluster nodes will have at least one integrated 100BaseT or better (e.g., 1000Base-SX) management Ethernet.

3.2.2.11 LinuxBIOS (TR-3)

The MCR cluster nodes will have LinuxBIOS implemented for each node type (see www.linuxbios.org).

3.2.2.12 QsNet Elan3 Network Integration (TR-1)

The Offeror shall install in the MCR P4 cluster nodes one Government Furnished Equipment (GFE) QsNet Elan3 QM400 Network adapter. This card is a PCI 66MHz/64-bit, PCI 2.1 Universal 3V/5V operation card. See URL: http://www.quadrics.com/website/pdf/qsnet.pdf.

3.2.3 Remote Manageability (TR-1)

All MCR nodes will be 100% remotely manageable. That is, all service operations on the node must be accomplished without the attachment of keyboard, video monitor and mouse (KVM). Any provided remote management tools, API's or protocols will have open, published specifications so that the University can use, modify or write Open Source management tools to utilize them.

The Offeror will fully describe all remote manageability features, protocols, APIs, utilities and description of management operations of every node type bid. Any available manuals (or URL's pointing to those manuals) describing node management procedures for each node type will be provided with the proposal.

3.2.3.1 Serial Console Redirection (TR-1)

All BIOS interactions will be through a serial console. There should be no system management operations on a node that require a graphics subsystem or keyboard, video or mouse (KVM) or CDROM or floppy to be plugged in. In particular, the serial console will display POST/failure codes, operate even upon failure of CMOS battery and provide for a mechanism to remotely access the BIOS.

3.2.3.2 Dedicated Serial Console Communications (TR-2)

The serial console communication channel will be available for console logging and interactive use at all times. It will not be shared with other devices such as embedded service processors. This is to ensure that all console output is logged and Expect scripts that perform console or service processor actions do not interfere with each other or with console logging.

3.2.3.3 Serial Console Efficiency (TR-1)

BIOS output should not repaint the screen (e.g., menu pages or memory check progress bars) in such a way as to slow the console output and thereby increase reboot time. It is desirable that the serial console will support a baud rate of 38400 or greater and RTS/CTS hardware flow control..

3.2.3.4 BIOS Command Line Interface (TR-2)

A scriptable command-line interface (CLI) to the BIOS will be provided. In order to facilitative scripting BIOS interactions with the Linux Expect utility, all interactions with the BIOS will use this CLI and will not require the navigation of ANSI menus.

3.2.3.5 CMOS Parameter Manipulation (TR-1)

The Offeror will provide a mechanism to read (get) and write (set) CMOS parameters from Linux command line (see section 3.3.3.1). In particular, CMOS parameters such as boot order will be modifiable from Linux command line. In addition, there will be a CMOS parameters that allows system disable any "green" power management features. The Linux command line utility will also allow reading all CMOS parameters (backup) and writing all CMOS parameters (restore). Reading or writing CMOS parameters will not require booting an alternative operating system or interacting with BIOS menus or BIOS CLI.

3.2.3.6 Failsafe CMOS Parameters (TR-1)

The BIOS will have failsafe default parameter settings so that the serial console interface will function if the CMOS values are unintentionally reset to the default values (e.g., the CMOS battery fails.).

3.2.3.7 BIOS Upgrade (TR-1)

Offeror will provide a hardware mechanism that allows provided Linux command line utility or utilities to update (flash) the BIOS image in flash memory and to verify the BIOS flash image (see section 3.3.3.2).

3.2.3.8 Peripheral Device Firmware (TR-2)

Offeror will provide Linux a utility or utilities for saving, restoring, verifying (including printing version number) firmware for any peripheral devices supplied.

3.2.3.9 Power-On Self Test (TR-2)

The POST will be comprehensive, detect hardware failures and identify the failing FRU during the power up boot sequence. All POST failures should be reported to the serial console or otherwise remotely accessible.

The requirements of this section, taken together provide mechanisms so that node BIOS upgrade or configuration changes will be accomplished without using a real or virtual floppy disk.

3.2.3.10 Remote Network Boot Mechanism (TR-1)

The node BIOS will support booting an executable image over the management Ethernet utilizing PXE, BOOTP or DHCP. See URL: `ftp://download.intel.com/ial/wfm/pxespec.pdf`.

3.2.4 Gateway Node Requirements (TR-1)

The following Requirements are specific to the thirty two gateway nodes and supercede the general node requirements, above. It is preferred that the gateway nodes be identical to the compute nodes.

3.2.4.1 Gateway Node Delivered PCI Performance (TR-1)

The MCR Gateway nodes and chip set will be configured to deliver at least 300 MB/s PCI 64b/66MHz bandwidth when running the MPI bandwidth benchmark across the Quadrics QsNet ELAN3 PCI-64b/66MHz adapter and simultaneously drive the two GFE NetGear GA621 1000Base-SX adapters at 180 MB/s with standard frames utilizing the TTCP benchmark. Note that this implies at least two independent PCI 64b/66 MHz buses. Offeror will report with proposal the Gateway node delivered MPI bandwidth benchmark over ELAN3 PCI 64b/66 MHz adapter performance and simultaneous aggregate delivered TTCP benchmark performance as a function of buffer size over two NetGear GA621 1000Base-SX adapters utilizing standard frames.

We recommend the "com" benchmark from the ASCI Purple Presta MPI Stress Test suite. Presta can be obtained from:
http://www.llnl.gov/asci/purple/benchmarks/limited/presta/

3.2.4.2 Gateway Node Configuration (TR-1)

Offeror will integrate one GFE Quadrics Elan3 QM400 host bus adapter. Offeror will install two GFE NetGear GA621 1000Base-SX Ethernet host bus adapters.

3.2.5 Meta Data Server Fail-Over Pair Node Requirements

The following Requirements are specific to the two Lustre Meta Data Server (MDS) nodes and supercede the general node requirements, above.

3.2.5.1 MDS Fail-Over Configuration (TR-1)

The two MDS nodes will be configured to operate as a Kimber Lite High Availability fail-over pair (http://oss.missioncriticallinux.com/projects/kimberlite/). That is, there will be a dedicated serial port link for heart beat between the two nodes. There will be dedicated (private) Ethernet link between the two nodes, in addition to the management Ethernet. The proposed disk subsystem for the MDS nodes shall be accessible from either node.

3.2.5.2 MDS Shared Meta Data Disk Space (TR-1)

In addition to the local disk on each MDS node, the MDS nodes will share 2.5 TB of RAID5 disk for Lustre Meta Data. Each MDS node will have access to the full 2.5 TB of Lustre Meta Data. The shared meta data disk space will have high availability characteristics such as no single point of failure and hot spare disks.

3.2.5.3 MDS Node Form Factor (TR-1)

The Offeror will provide MDS nodes packaged in standard 19" rack mountable enclosures with at most 4U form factor per node. A smaller form factor that does not limit the number or type of PCI adapters that can be installed is preferred.

3.2.5.4 MDS Node Delivered PCI-X Performance (TR-1)

The MCR MDS nodes and chip set will be configured to deliver at least 300 MB/s PCI 64b/66 MHz bandwidth when running the MPI bandwidth benchmark across the Quadrics QsNet ELAN3 PCI 64b/66 MHz adapter and simultaneously drive raw disk. The raw disk I/O devices will be attached to PCI-X and deliver 256 MB/s with 512B block raw read/write (or 500x512B I/O's per second) performance from either node. Note that this implies at least two independent PCI-X buses. Offeror will report with proposal the MDS node delivered MPI bandwidth benchmark over ELAN3 PCI 64b/66 MHz adapter performance and simultaneous delivered raw disk I/O over PCI-X performance.

3.2.5.5 MDS Node Configuration (TR-1)

Offeror will integrate one GFE Quadrics Elan3 QM400 host bus adapter.

3.2.6 Login Node Requirements

The following Requirements are specific to the Login nodes and supercede the general node requirements, above.

3.2.6.1 Login Node Form Factor (TR-1)

The Offeror will provide login nodes packaged in standard 19" rack mountable enclosures with at most 4U form factor per node. A smaller form factor that does not limit the number or type of PCI adapters that can be installed is preferred.

3.2.6.2 Login Node Memory Size (TR-1)

The login nodes will be configured with at least 4.0 GiB (2.0 GiB of memory per processor).

3.2.6.3 Login Node Delivered PCI Performance (TR-1)

The MCR Login nodes and chip set will be configured to deliver at least 300 MB/s PCI 64b/66 MHz bandwidth when running the MPI bandwidth benchmark across the Quadrics QsNet ELAN3 PCI 64b/66 MHz adapter and simultaneously drive four NetGear GA621 1000Base-SX Ethernet PCI 64b/66 MHz adapters with Jumbo Frames attached to PCI 64b/66 MHz at an aggregate of 256 MB/s with a single four way parallel FTP or four copies of serial FTP. Note that this implies at least two independent PCI 64b/66 MHz buses. Offeror will report with proposal the Login node delivered MPI bandwidth benchmark over ELAN3 PCI 64b/66 MHz adapter performance and simultaneous delivered FTP over GA621 1000Base-SX in PCI 64b/66 MHz performance.

We recommend the "com" benchmark from the ASCI Purple Presta MPI Stress Test suite. Presta can be obtained from:
http://www.llnl.gov/asci/purple/benchmarks/limited/presta/

3.2.6.4 Login Node Configuration (TR-1)

Offeror will integrate one GFE Quadrics Elan3 QM400 host bus adapter. The Offeror will provide and integrate four (4) NetGear GA621 1000Base-SX Ethernet or equivalent cards. Two will be configured for Jumbo frame and two will be configured for normal frame usage.

3.2.7 MCR Scalable Unit (TR-1)

It is essential that an MCR scalable unit is defined by the Offeror in the proposal. These scalable units will minimize footprint, unused space, and include a group of Nx96 nodes and at least a group of NxQsNet level one switches. N will be 1, 2, 3, 4 or 5. This construct will also facilitate building and testing MCR prior to shipment to LLNL. The following example is given for ease of discussion.

The following example assumes 96 node scalable units with one QsNet level one switch per scalable unit. The second level switches will be housed in separate racks physically located in the center of the other scalable units. Two scalable units are defined: First Scalable Unit FSU and Compute Node Scalable Unit (CNSU). The FSU will be constructed first and connected to the lowest QsNet ports in the federated switch. After the FSU there are nine identical CNSU.. It is assumed that that at least 21x2U compute nodes can fit into a standard 19" 42U rack and that one rack is 47U. It is assumed that 10x4U I/O or Login nodes can be placed into a standard 19" 42U rack. If the proposed compute node form factor is less than 2U or the I/O or Login node form factor is less than 4U, then the Offeror will propose alternate scalable units that have sufficient 1.5U Ethernet management switches and Quadrics 17U switches in a densely packed set of racks. If alternate scalable units are proposed, the response to the requirements below should reflect the proposed modifications, but should follow the same outline.

3.2.7.1 Node Racks (TR-1)

The Offeror will place the P4 compute nodes in standard 19" racks with ample room for cable management of QsNet ELAN3 cables, CAT5 Ethernet cables and console serial cables and power cables. There shall not be any console display, keyboard or mouse in the rack.

3.2.7.2 First Scalable Unit (TR-1)

Offeror will supply at least one (1) First Scalable Unit (FSU). The FSU will be configured as in Figure 3.2-1. The FSU will be the first scalable unit built and configured in the lowest ports on the federated QsNet switch. The FSU will contain 96 nodes on the first first level QsNet switch in six 19"x42U racks and 98 nodes on the management Ethernet. These nodes will be 2x4U Login nodes, 2x4U Lustre Meta Data (fail-over pair) nodes, 32x2U gateway nodes and 60x2U compute nodes. This FSU will also contain one 17U GFE QsNet Elan3 switch and two management nodes (not connected to the QsNet switch). The FSU will also contain 3x1.5U Cisco management Ethernet switches and all nodes in the FSU will be connected to the management Ethernet.

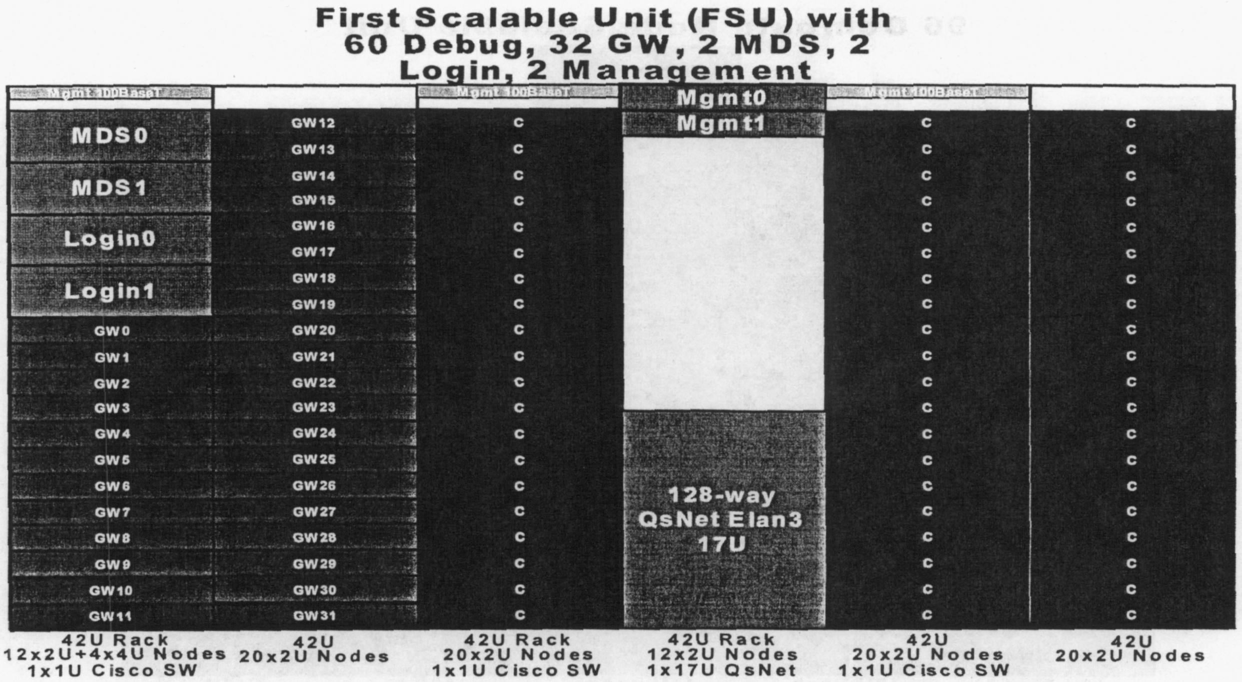


Figure 3.2-1 First Scalable Unit contains one 17U QsNet Elan3 switch, 2x4U meta data servers, 2x4U login nodes, 32x2U gateways, 60x2U compute nodes configured as an interactive debug partition and 3x1.5U Cisco management Ethernet switches in six 42U racks.

3.2.7.3 Compute Node Scalable Unit (TR-1)

Offeror will supply at least nine (9) Compute Node Scalable Units (CNSU). The CNSU will be configured as in Figure 3.2-2: four (4) 42U racks with 82x2U compute nodes (2x21+2x20) and 2x1.5U Cisco management Ethernet switches ; one (1) 47U rack with 14x2U compute nodes and 1x17U Quadrics QsNet Elan3 switch.

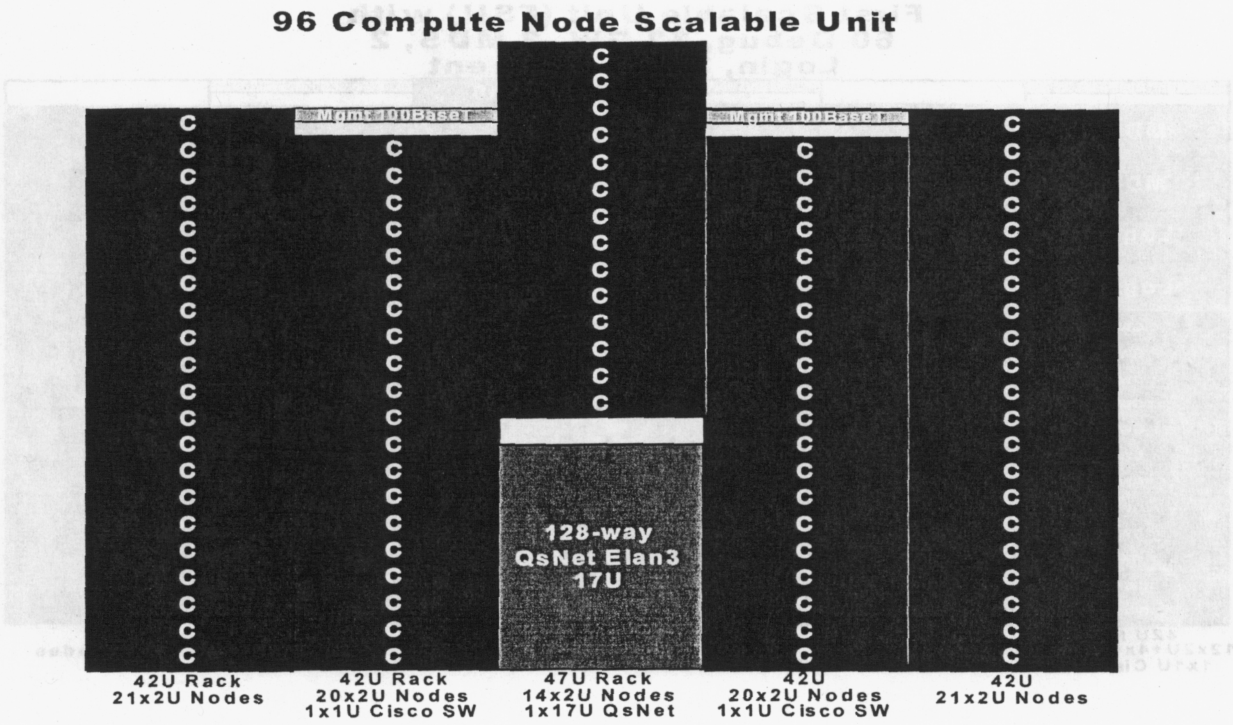


Figure 3.2-2: 96 node Compute Node Scalable Unit contains one 17U QsNet Elan3 switch, 96x2U form factor compute nodes, 2x1.5U Cisco management Ethernet switches in four 42U racks and one 47U rack.

3.2.7.4 MCR Management Ethernet Switch (TR-1)

The Offeror will provide 1.5U Cisco Catalyst 3548-XL-EN 10/100/1000 48 port Ethernet switch for the MCR management Ethernet. The management Ethernet cables will be bundled within the rack in such a way as to not kink the cables, nor place strain on the Ethernet connectors. All Management Ethernet connectors will have a snug fit when inserted in the Management Ethernet port on the nodes. The management Ethernet cables will meet or exceed Cat 5E specifications for cable and connectors. Cable quality references can be found at: (http://www.integrityscs.com/index.htm) and (http://www.panduitncg.com:80/whatsnew/integrity_white_paper.asp).
A suggested source of this quality cable is Panduit corporations Powersum+ tangle free patch cords, Part# UTPCI10BL for a 10' cable. The URL for this product is: (http://www.panduitncg.com/solutions/copper_category_5e_5_3.asp).
Management Ethernet reliability is specified in section 4.1.

3.2.7.5 MCR Rack SPC and RPC (TR-1)

The Offeror shall provide sufficient Serial Port Concentrators (SPC) and Remote Power Controller RPC) for each node rack. The specifications for these SPCs and RPCs are defined below.

An example of a preferred combined SPC and RPC solution is the Linux NetworX IceBox. The IceBox solution does not require vertical rack space and does SPC and RPC for ten (10) outlets at 30 amps. See (http://www.linuxnetworx.com/products/icebox.php).

3.2.7.5.1 Serial Port Concentrators (TR-1)

Each MCR rack will be equipped with Serial Port Concentrators (SPC) for node console serial port and other management hardware, if applicable. The serial console port of every node in each MCR rack will interface to serial port concentrator(s) in that rack. The SPCs will interface to the MCR management Ethernet. All serial console traffic from every node attached to an SPC will be bridged to the management Ethernet via telnet protocol (RFC 854). The SPC will operate node serial ports at 38400 baud or greater with RTS/CTS hardware control flow.

3.2.7.5.2 Remote Power Control (TR-1)

Each MCR rack will be equipped with Remote Power Controler and concentrator (RPC) for node power management. The RPC will combine node power cords so that two or four higher amperage cords are required to site the rack. Power cables from nodes to RPC will be provided. The node power cables will fit snugly into both end receptacles and will be of sufficient quality to not become a fire hazard during the 3yr cluster lifetime. Power cables from RPC to LLNL power receptacles will be provided (indicate in siting section 5.3 where receptacles should be placed). The RPC will allow, over the management Ethernet via telnet protocol (RFC 854), system administrators to power on nodes, power off nodes, reset nodes and reliably determine the current power status of nodes in the rack. It will not be possible to power down the node in such a way that it cannot be powered up again by the proposed power management solution.

3.2.7.5.3 SPC and RPC Protocol Specification (TR-1)

Offeror will fully specify the protocol used by the SPC and RPC devices in the response. This documentation will be sufficiently detailed to allow the University to evaluate how existing or in-house developed Open Source Software could manage these devices. Any protocol specification copyright, if applicable, will not inhibit the Open Source distribution of such University developed management software.

3.2.7.5.4 SPC Security (TR-2)

Access control lists or equivalent will provide a method to prevent unauthorized access to SPC and RPC from the management Ethernet.

3.2.7.5.5 SPC and RPC Zero Vertical Space (TR-1)

The combination of SPC and RPC solutions will not require any vertical space in any of the MCR racks. That is, the SPC and RPC solutions will mount in the MCR racks so that all 42U of space is available for nodes and/or management Ethernet switch.

3.2.7.6 MCR Rack Cooling (TR-1)

The Offeror will ensure the MCR racks and cabling do not inhibit the airflow into and out of the compute nodes. Additional fan units, if necessary, may mount to the rear door and swing away when servicing the back of the rack.

3.3 MCR Software Requirements

This section describes the software requirements beyond the GFS for the MCR. The software associated with building and installing the MCR is described in sections 2.2, 6.3.4, 6.3.5, 6.3.8.

3.3.1 MCR Peripheral Device Drivers (TR-1)

Offeror will provide Linux drivers for all peripheral devices supplied (i.e., peripheral hardware bid beyond the GFE) that function with the CHAOS kernel. Offeror will specifically call out and fully disclose any proposed peripheral device drives required with the proposed system including version number and provide system administration or programmers documentation with the response.

3.3.2 Memory Error Kernel Module (TR-2)

The Offeror will provide a memory error kernel module for Linux 2.4.X releases that interfaces to the memory SECDED hardware (section 3.2.2.7). This kernel module will log all single and double bit memory errors to the Linux kernel log facility and report the offending memory RIMM or DIMM (or other lowest level of memory FRU). This kernel module will panic the node if the memory subsystem generates a double bit memory error. This kernel module will provide an appropriate interface to the hardware diagnostics in section 4.5.2.

3.3.3 Remote Management Software (TR-2)

The Offeror will provide remote management software, beyond that defined in section 3.3.1, for the remote management of the MCR Cluster. This may include utilities to capture and monitor BIOS, Linux Console and other node management I/O. It is preferred that any provided software is Open Source.

3.3.3.1 Node CMOS Parameter Manipulation (TR-1)

The Offeror will provide a utility or utilities to implement the CMOS Parameter Manipulation hardware requirement section 3.2.3.5.(

3.3.3.2 Node BIOS Upgrade (TR-1)

Offeror will provide a Linux command line utility or utilities that update (flash) the BIOS image in flash memory and to verify the BIOS flash image (see section 3.2.3.7). This may require a patch to the memory technology device (MTD) driver in the Linux kernel (www.linux-mtd.infradead.org). In particular, flashing or verifying the BIOS will not require booting an alternative operating system or interacting with BIOS menus or BIOS CLI.

3.3.3.3 Environmental Sensors (TR-1)

Support for reading any environmental sensors on the motherboard via the "lm_sensors" package will be provided. This includes any kernel driver support (e.g. I2C and sensor devices) and appropriate configuration files for threshold values. See http://www.netroedge.com/~lm78

3.3.4 System Diagnostics (TR-2)

See section 4 for the list of system monitoring and diagnostics required.

End of Section 3

4 Reliability, Availability, Serviceability (RAS) and Maintenance

The MCR cluster is intended for production usage in the M&IC Open Computing facility. Thus, the University requires that the cluster have highly effective, scalable RAS features and prompt hardware maintenance.

For hardware maintenance, the strategy is that University personnel will provide on-site, on-call 24x7 hardware failure response. We envision that these hardware technicians and system administrators will be trained by the Offeror to perform on-site service on the provided hardware. For easily diagnosable node problems, University personnel will perform repair actions in-sutu by replacing Field Replaceable Units (FRUs). For harder to diagnose problems, University personnel will swap out the failing node(s) with on-site hot spare node(s) and perform diagnosis and repair actions in the separate Hot-Spare Cluster (HSC). Failing FRUs or nodes will be returned to the Offeror for replacement. Thus, the University requires an on-site parts cache of all FRUs and a small cluster of fully functional hot-spare nodes of each node type. The Offeror will work with the University to diagnose hardware problems (either remotely or on-site, as appropriate). On occasions, when systematic problems with the cluster are found, the Offeror's personnel will augment University personnel in diagnosing the problem and performing repair actions.

In order for the very large MCR cluster to fulfill the mission of providing the capability resource for M&IC, it must be stable from both a hardware and software perspective. Thus, the number of failing components per unit time (weekly) should be kept to a minimum. System components should be fully tested and burned in before delivery (initially and as FRU or hot-spare node replacement). In addition, in order to minimize the impact of failing parts, the University must have the ability to quickly diagnose problems and perform repair actions. Thus, a comprehensive set of diagnostics that are actually capable of exposing and diagnosing problems are required. It has been our experience that this is a difficult, but achievable goal and the Offeror will need to specifically apply sufficient resources to accomplish it.

Through the LLNL "HotLine" facility, hardware and software problems will be reported and tracked to solution. Our software strategy is similar to the hardware strategy in that University personnel will diagnose software bugs to determine the failing component. The problem will be handed off to the appropriate organization for resolution. For University supplied system tools, University personnel will fix the bugs. For Offeror supplied system tools, the Offeror will need to supply problem resolution. For the Linux kernel and associated utilities, the University will separately contract with RedHat for Enterprise level support. For QsNet related HW/SW problems, the University will separately contract with Quadrics for support. For compilers, debugger and application performance analysis tools, the University will separately contract with the appropriate vendors for support.

4.1 Highly Reliable Management Network (TR-1)

The management Ethernet will be a highly reliable network that does not drop a single node from the network more than once a month. That is, the connection between any MCR node and the management network will be dropped less than once every 920 months. This is both a hardware and a software (Linux Ethernet device driver) requirement. In addition, the management network will be implemented with connectors on the node mating to the management Ethernet cabling and connectors (section 3.2.7.4) so that manually tugging or touching the cable at a node or switch does not drop the Ethernet link. The Cisco management Ethernet switches (section 3.2.7.4) will be configured such that they behave as standard multi-port bridges.

4.2 MCR Node Reliability and Monitoring (TR-2)

The MCR nodes will have high reliability characteristics such as redundant fans and power supplies. The MCR will have a real-time hardware monitoring, at a specified interval, of system temperature, processor temperature, fan rotation rate, power supply voltages, etc. This node hardware monitoring facility will alert the scalable monitoring software in section 4.6 via serial console or management network when any monitored hardware parameter falls outside of the specified nominal range. In addition, the system components may provide failure or diagnostic information via the serial console or management network.

4.3 In Place Node Service (TR-1)

The nodes will be serviceable from within the rack. The node will be mechanically designed to have minimal tool requirements for disassembly. The nodes will be mechanically designed so that complete node disassembly and reassembly can be accomplished in less than 20 minutes by a trained technician.

4.4 Component Labeling (TR-1)

Every rack, Ethernet switch, Ethernet cable, ELAN3 switch, ELAN3 cable, node, disk enclosure will be clearly labeled with a unique identifier visible from the front of the rack and/or the rear of the rack, as appropriate, when the rack door is open. These labels will be high quality so that they don't fall off, fade or disintegrate or otherwise become unusable or unreadable during the lifetime of the cluster. Nodes will be labeled from the rear with a unique serial number for inventory tracking. It is desirable that motherboards also have a unique serial number for inventory tracking. This serial number needs to be visible without having to disassemble the node, or else it must be queryable, either from Linux or the BIOS.

4.5 Field Replaceable Unit (FRU) Diagnostics (TR-2)

Diagnostics will be provided that, at a minimum, isolates a failure of a MCR cluster component at LLNL to a single Field Replaceable Unit (FRU) for the supplied hardware. These diagnostics will run from the Linux command line. The diagnostic information will be accessible to operators through networked workstations.

4.5.1 Node Diagnostics Suite (TR-1)

The Offeror will provide a set of hardware diagnostic programs (a diagnostic suite or diagnostics) for each type of node provided that run from the Linux command line and produce output to STDERR or STDOUT and exit with an appropriate error code when errors are detected. These diagnostics will be capable of stressing the node motherboard components such as processors, chip set, memory hierarchy, on-board networking (e.g., management network), peripheral buses and local disk drives. The diagnostics will stress the memory hierarchy to generate single and double bit memory errors. The diagnostics will read the hardware single and double bit memory error counters and resetting the counts to zero. The diagnostics will stress the local disk in a non-destructive test to generate correctable and uncorrectable read and/or write errors. The diagnostics will read the hardware and/or Linux recoverable I/O error counters and resetting the counts to zero. The diagnostics will stress the integer and floating point units in specific processor(s) in serial (i.e., one processor and/or HyperThread, as appropriate, at a time) or in parallel (i.e., multiple processors and/or multiple HyperThreads, as appropriate). The CPU stress tests will bind to a specific processor and/or HyperThread, as appropriate, by command line option, if possible.

4.5.2 Memory Diagnostics (TR-1)

The provided Linux OS will interface to the hardware memory SECDED facility specified in section 3.2.2.7 to log all single bit and double bit errors on each memory FRU. When the node experiences a double bit memory error, the Linux kernel will report the failing memory FRU and panic the node. The Offeror will provide a Linux utility that can scan the nodes and report single bit and double bit memory errors at the FRU level and reset the counters.

4.5.3 Peripheral Component Diagnostics (TR-2)

The Offeror will provide a set of hardware diagnostic programs (a diagnostic suite or diagnostics) for each type of peripheral component provided that run from the Linux command line and produce output to STDERR or STDOUT and exit with an appropriate error code when errors are detected. At a minimum, the diagnostics will test the 1000Base-ST and other provided networking (e.g., Fibre Channel) adapters, RAID device and disks.

4.6 Scaleable System Monitoring (TR-2)

There will be a scalable system monitoring capability for the MCR cluster that has a command line interface (CLI) for scriptable control and monitoring. This facility will directly interface to the node monitoring facility through the serial console or management network and control the node monitoring and diagnostics facilities. All system monitoring information will be centrally collected at intervals set by the system administrator on one of the service nodes. All system monitoring and diagnostics information will be formatted so that “expect scripts” can detect failures. In addition, this facility will be able to launch diagnostics in parallel over the management network across all or part of the cluster, as directed by a system administrator from the Linux command line on one of the service nodes.

4.7 Hardware Maintenance (TR-1)

The Offeror will supply hardware maintenance for provided components of the proposed MCR system for a three-year utilization period. University personnel will attempt on-site first-level hardware fault diagnosis and repair actions. Offeror will provide second-level hardware fault diagnosis and fault determination during normal business hours. That is, if the University cannot repair failing components based on-site parts cache, then the Offeror personnel will be required to make on-site repairs. Offeror supplied hardware maintenance response time will be before the end of the next business day from incident report until Offeror personnel perform diagnosis and/or repair work. The proposed system will be installed in an a Property Protected Area at the Laboratory and so maintenance personnel will obtain DOE P clearances.

4.7.1 On-site Parts Cache (TR-1)

A parts cache (of FRUs and hot spare nodes of each type proposed) at LLNL is required that will be sufficient to sustain necessary repair actions on all proposed hardware and keep them in fully operational status for at least one week without parts cache refresh. That is, the parts cache, based on Offeror's MTBF estimates for each FRU and the entire system, will be sufficient to perform all required repair actions for one week without the need for parts replacement. The Offeror will propose sufficient quantities of FRUs and hot-spare nodes for the parts cache. The parts cache will be enlarged, at the Offeror's expense, should the on-site parts cache prove in actual experience to be insufficient to sustain the actually observed FRU or node failure rates. However, at a minimum, the on-site parts cache will include the following fully configured (except for QsNet Elan3 adapter) nodes: ten compute nodes, one login node (if they are different from either compute or gateway nodes). The Offeror will supply sufficient racks and associated hardware/software to make the HSC a cluster that can run diagnostics on every HSC node over the management Ethernet. In addition, a minimum of twenty of the following parts, if bid: local disk drives, RDRAM RIMM kit for a node, SDRAM DIMM kit for a node, power supplies of each type and fans of each type. One each of the following, if bid: Linux Networx Ice Box (or equivalent), Cisco Ethernet switch, any other replicated rack component other than nodes or cables. The University will administer the nodes as a separate hot spares cluster in the unclassified environment. The University will store and inventory the HSC and other on-site parts cache components.

4.8 Mean Time Between Failure (MTBF) Calculation

The Offeror will provide the Mean Time Between Failure (MTBF) calculation for each FRU and node type. The Offeror will use these statistics to calculate the MTBF for the provided aggregate MCR cluster hardware. This calculation will be performed using a recognized standard. Examples of such standards are Military Standard (Mil Std) 756, Reliability Modeling and Prediction, which can be found in Military Handbook 217F, and the Sum of Parts Method outlined in Bellcore Technical Reference Manual 332. In the absence of relevant technical information in the proposal, the University is forced to make pessimistic reliability, availability and serviceability assumptions in evaluating the proposal.

End of Section 4

5 Facilities Information

A portion of an existing facility, the main computer floor of B439, will be used for siting the MCR system. This entire facility has approximately 8,000-9,000 ft^2 and 1.9 MW of power for the computing system and peripherals and associated cooling available for this purpose in B439. The computer floor is 24" raised floor with 150 lbs per square inch loading capability. Power will be provided to racks by under floor electrical outlets supplied by the University to Offeror's specifications. Circuit breakers and PDUs are available in wall panels that can be modified to Offeror's specifications. All other cables can be laid on the floor over the electrical conduit. Straight point-to-point cable runs can be assumed. The University will provide floor tile cut to Offeror's specifications. In addition, it is anticipated that the Offeror's equipment will be placed in adjacent rows so that air intakes in racks from adjacent rows are abutting with Offeror's specified separations and hot air exhausts in racks from adjacent rack rows are abutting with Offeror's specified separations. That is, the racks will be placed so that there are HOT and COLD isle ways between racks with chilled air entering in the COLD isles and warmed air exiting in the HOT isles.

Thus, it is essential the Offeror make available to the University detailed and **accurate** (not grossly conservative over estimates) site requirements for the MCR system as bid (not fully loaded) at proposal submission time.

5.1 Power Requirements (TR-1)

Power requirements will be fully disclosed by Offeror at the time of proposal submission. This information will be communicated in the Offeror's response. Offeror will provide for each rack type: the number of KW or KVA required, the number and type of power connections required and anticipated electrical load. This information will be verified by joint written (e-mail and text files) and telecons between the Offeror after contract award and the University within three weeks of contract award.

5.2 Cooling Requirements (TR-1)

Cooling requirements will be fully disclosed by Offeror at the time of proposal submission. This information will be communicated in the Offeror's response. Offeror will provide for each rack type: the number of BTU or Tons AC required, any environmental requirements: such as temperature and/or humidity range requirements. This information will be verified by joint written (e-mail and text files) and telecons between the Offeror after contract award and the University within three weeks of contract award.

5.3 Floor Space Requirements and Floor Plan (TR-1)

The following example shows a high level diagram of how the MCR could be laid out, given the FSU and CNSU as described in section 3.2.7.

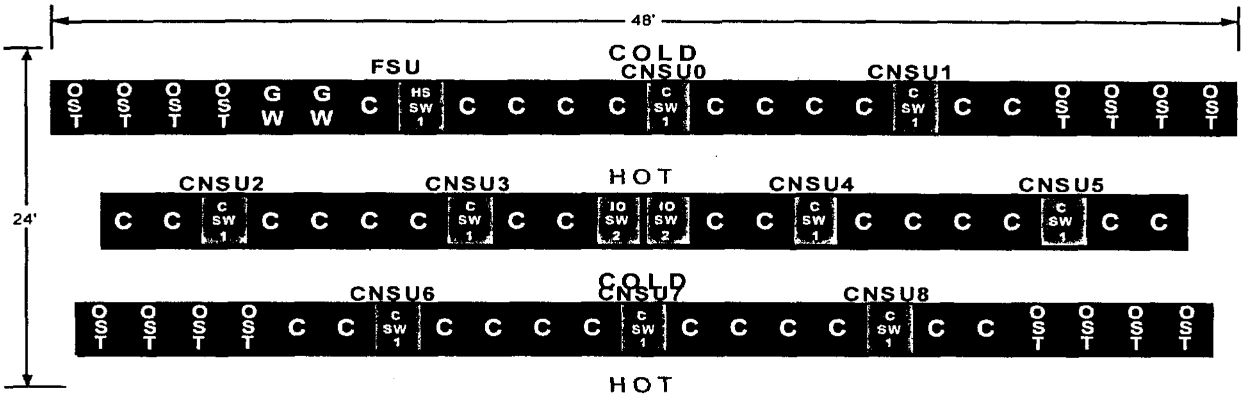

48'
COLD
FSU
CNSU0
CNSU1
OST OST OST OST GW GW C HS SW 1 C C C C C SW 1 C C C C C SW 1 C C OST OST OST OST
CNSU2
CNSU3
HOT
CNSU4
CNSU5
24'
C C C SW 1 C C C C C SW 1 C C IO SW 2 IO SW 2 C C C SW 1 C C C C C SW 1 C C
CNSU6
COLD
CNSU7
CNSU8
OST OST OST OST C C C SW 1 C C C C C SW 1 C C C C C SW 1 C C OST OST OST OST
HOT

Figure 5.3-1 MCR layout with the example FSU, CNSU and OST GFE can be laid out with a floor plan of 24'x48'.

Floor space requirements will be fully disclosed by Offeror at the time of proposal submission. This information will be communicated in the Offeror's response. Offeror will provide a detailed floor plan (system layout) diagram indicating rack placement and location of required electrical outlets. This information will be verified by joint written (e-mail and text files) and telecons between the Offeror after contract award and the University within three weeks of contract award.

5.4 Delivery Requirements (TR-1)

If Offeror has any delivery requirements these will be communicated to the University in the proposal. The MCR cluster will reside in a Property Protected Area and installation personnel will be US Citizens or Aliens with Green Cards from Tier one or Tier two countries. All installation personnel will obtain a DOE P clearance for site access. Unless otherwise indicated in Offeror's response, installation crews will work up to an eight (8) hour day Monday through Friday 8:00AM to 5:00PM. Longer days, differing shift start/end times and/or weekend shifts can be accommodated by the University at Offeror's request at least one week prior to delivery.

End of Section 5

6 Project Management

The construction, ship testing, delivery, installation, acceptance testing of the MCR clusters is a complex and non-trivial endeavor. It is anticipated that this project will require close coordination of University, Quadrics and the Offeror's personnel.

6.1 Open Source Development Partnership (TR-2)

The Offeror will provide information on the capabilities of the corporation to engage in an Open Source development partnership and meet the goals set out in section 1.7. This information should include corporations financial health; corporation's qualifications as a cluster provider; corporation's qualifications as an Open Source development organization; cluster product roadmap and comparison to the overall PCR strategy; the willingness of the corporation to participate in the Open Source development, with other partners, of key missing HPTC cluster technology components such as scalable parallel file systems and cluster resource scheduling. If the Offeror has technology, such as a scalable parallel file system or cluster management tools or cluster resource scheduling that could be contributed to the Open Source community, please indicate that as well.

6.2 Project Manager (TR-1)

The Offeror will provide the name and resume of proposed project manager within the Offeror's corporation for the proposed activity. This project manager will be approved by the University's technical representative. The project manager will be empowered by the Offeror's corporation to plan and execute the construction, shipment and installation of the proposed configuration. This will include sufficient personnel and hardware resources within the corporation to assure successful completion of the activity on the proposed schedule.

6.3 Project Milestones

The delivery of the proposed hardware must be accomplished before the end of September 2002. The University plans to transition the MCR clusters to unclassified operation for full-system science runs within thirty (30) days of delivery. The University plans to run full-system science runs for approximately four months and then transition the MCR cluster to limited availability (LA) for a small group of M&IC users. The University plans to migrate MCR to general availability (GA) status within thirty (30) days of LA status. In addition, there is Government Furnished Equipment (GFE) and Government Furnished Software (GFS) that must be coordinated. In order to assure the timely execution of these programmatic goals and to make sure both parties understand the timeline, the Offeror will provide the University with a project plan within 7 days of contract award.

6.3.1 Detailed Project Plan (TR-1)

The Offeror will provide a detailed project plan 7 days after contract award. This project plan will include a Gantt chart with all the project milestones with dates and durations for work activities leading up to the milestones. The Gantt chart will indicate work activity and milestone and organizational dependencies. The Gantt chart will clearly indicate the projects critical path. At least one level of detail below each of the project milestones showing the work activities leading up to completion of the milestone will be included in the Gantt chart. The project plan will include a written pre-ship test plan and acceptance test plan. The project plan Gantt chart will be a Microsoft Project 2000 data file. The test plans will be

Microsoft Word 2000 data files. This milestone is complete when the University approves the project plan.

6.3.2 Early Node Delivery (TR-2)

The Offeror may deliver the at least one node of each type up to the full hot spare pool (section 4.7.1) quickly after contract award. In addition, one each of the SPC and RPC solutions (see sections 3.2.7.5) may be provided quickly after contract award. This is to provide the University with access to the node technology to work out MCR software management compatibility and to have an environment with which to build the boot disk image. Offeror will indicate early node delivery date in proposal. This milestone is complete when the nodes are installed on-site at LLNL, boot up and run the single node threaded LINPACK benchmark successfully for four hours without failure or wrong answers.

6.3.3 GFE QsNet Elan3 Equipment (TR-1)

The University will provide QsNet switches, cables and adapters to Offeror at least four weeks before ship date. Quadrics technical personnel will be on-site for up to two weeks at Offeror's cluster build location to help with QsNet installation, stabilization and pre-ship test execution. Please indicate when and where this equipment is required in the proposal response. Please indicate when and where the Quadrics technical personnel should travel for QsNet installation, testing and pre-ship test execution in the proposal response. This milestone is complete when Offeror acknowledges that all the QsNet Elan3 hardware is at the build location.

6.3.4 Linux Build Image (TR-1)

The Offeror will work with Quadrics and Cluster File Systems, Inc. (CFS) to determine the Linux based disk image (software set) to install on the MCR cluster during the MCR build process. The Offeror will utilize Offeror's own Linux distribution and (possibly proprietary) tools for this installation, burn-in and pre-ship test. The University's technical representative will approve the Linux Build Image prior to build of the FSU, but it must include the Quadrics RMS, QsNet device driver and MPI stack as well as a functional remote console and power management solution, the Kimber Lite High Availability Cluster Infrastructure, Lustre Lite file system and associated kernel modifications and software components (e.g., OpenLDAP, SNMP, XFS). Please indicate in the response when this image is required for installation on FSU in the proposal response. This milestone is complete when: 1) the University's technical representative is able to confirm with Offeror, Quadrics and CFS all agree that the proposed software set will produce a functional FSU and MCR; 2) the software set installation is confirmed by running LLNL's Prest MPI bandwidth test (com, see http://www.llnl.gov/asci/purple/benchmarks/limited/presta/presta.readme.html) on at least two nodes interconnected with Elan3.

6.3.5 FSU Build (TR-1)

The Offeror will build the First Scalable Unit (section 3.2.7.2), as bid. However, the FSU will include at least the ninety-six nodes (at least two Login nodes, at least two (fail-over) MDS nodes with at least 2.5 TB of shared disk for meta data, at least thirty-two gateway nodes, at least sixty compute nodes, configured in a debug partition) attached to initial FSU GFE (sixteen BlueArc OST via 1000Base-SW Ethernet and ninety-six Quadrics Elan3 HBA and switch). In addition, the FSU will contain the two management nodes (not on the QsNet switch). This milestone is complete when: 1) all FSU hardware is installed, burned-in and functional (all nodes must be functional, management Ethernet must be functional); 2) the University's technical representative confirms that Linux Build Image software set is installed on all FSU nodes; 3) LLNL's Presta MPI tests run for four hours complete successfully without any performance anomalies; 4) Lustre Lite file system is created and 15 TB of data is written and read without error including MDS fail-over test; 5) remote power and console management demonstrates console traffic and power cycling all the nodes in FSU.

6.3.6 Shipment Criteria (TR-1)

The Offeror will assemble and test the bid MCR cluster before shipment. Assembly will include installation of GFE on the MCR cluster, as bid. Quadrics technical personnel will be on-site at Offeror's location for up to two weeks to help with QsNet installation, stabilization and acceptance test execution. The University's technical representative will authorize shipment based on the successful completion of the pre-ship test. The pre-ship test plan will be mutually agreeable, but will include:

1) successfully running with correct results three mixed MPI/OpenMP jobs (sPPM, UMT2K, LINPACK) sequentially or simultaneously across 90% of the 924 compute nodes for at least four hours without failure;
2) successfully running the LLNL Presta MPI stress test sequentially or simultaneously across 90% of the 924 compute nodes for four hours without failure or performance anomalies.
3) a demonstration that the Management Ethernet is functional, stable and reliable.

Offeror will be responsible for LINPACK tuning and execution. The University will be responsible for sPPM and UMT2K tuning and execution. Offeror will provide a draft pre-ship test plan with the response. The pre-ship test plan draft will include clear test entry and exit criteria as well as a list of testing activities and benchmarks. This milestone is complete when the pre-ship test plan exit criteria are met, the University's technical representative authorizes shipment and the equipment leaves the Offeror's site.

6.3.7 Delivery and Installation (TR-1)

Offeror will deliver and install the bid MCR cluster and on-site parts cache in B439. Quadrics technical personnel will be on-site at LLNL for up to two weeks to help with QsNet installation, stabilization and acceptance test execution. Offeror will provide a draft post-ship test plan with the response. The post-ship test plan draft will include clear test entry and exit criteria as well as a list of testing activities and benchmarks. The post-ship test plan will be mutually agreeable, but should be similar to the pre-ship test. This milestone is complete when the MCR cluster and on-site parts cache, as bid, are fully installed at LLNL with final GFE (all the hardware is on-site, accounted for and assembled into an integrated and functioning cluster) and the post-ship test plan exit criteria are met and the MCR is turned over to the University for configuration prior to Acceptance Testing.

6.3.8 LLNL Linux Build Image Installation (TR-1)

The Offeror will work with University personnel to integrate any hardware specific features (e.g., BIOS flash support, memory error kernel module) and modifications to CHAOS for proposed SPC and RPC solutions. The University will install, with the aid of Offeror personnel, the LLNL Linux Build Image on the MCR cluster prior to acceptance testing.

This milestone is complete when the MCR meets the acceptance test entry criteria (see section 6.3.9).

6.3.9 Acceptance Testing (TR-1)

Offeror will provide a draft acceptance test plan with the response. The acceptance test plan draft will include clear test entry and exit criteria as well as a list of testing activities and benchmarks. The acceptance test plan will be mutually agreeable, but should be similar to the pre-ship test. As part of the acceptance test, the Offeror and University will repeat the pre-ship test to verify the system is functional. Offeror will be responsible for LINPACK tuning and execution. The University will be responsible for sPPM and UMT2K tuning and execution. In addition, I/O testing to the global file system and external networking devices will be accomplished. The MCR clusters, as bid, will function according to this SOW as part of the exit criteria of the acceptance test. This milestone is complete when the acceptance test plan exit criteria are met.

End of Section 6

7 Appendix A Glossary

7.1 General

Mandatory requirements designated as (MR)	Mandatory requirements are items that are essential to the University requirements and reflect the minimum qualifications an Offeror must meet in order to have their proposal evaluated further for selection.
Mandatory Option requirements designated as (MO)	Mandatory Option requirements deal with features, components, performance characteristics, or upgrades whose availability as an option is deemed a Mandatory Requirement by the University. Hence, a proposal not meeting a Mandatory Option will be deemed technically nonresponsive. Because the University will variously elect to include or exclude such options in resulting orders, each should appear as a separately identifiable item in the "Alternate Proposals and Options" and "Price Proposal".
Target Requirements designated as (TR-1, TR-2 and TR-3)	Each paragraph so labeled deals with features, components, performance characteristics or other properties that is considered a part of the ASCI system but will not be a determining factor of response compliance. Target requirements are prioritized by a dash number, TR-1 being the most important. Taken together the aggregate of the MR, MO and TR-1 requirements form a baseline system. TR-1 targets are as important to the program as mandatory requirements, but not meeting any particular TR-1 target requirement is insufficient to render a proposal as non-responsive. TR-2 targets are second priority after TR-1 requirements. TR-2 requirements are considered goals that boost a minimal baseline system, taken together as an aggregate of MR, MO, TR-1 and TR-2 requirements, into the moderately useful category. TR-3 targets are third priority after TR-2 requirements. TR-3 requirements are considered stretch goals that boost a moderately useful system, taken together as an aggregate of MR, MO, TR-1, TR-2 and TR-3 requirements, into the highly useful category. Thus, the ideal MCR systems will meet or exceed all MR, MO, TR-1, TR-2 and TR-3 requirements. Target Requirement responses will be considered as part of the evaluation of Technical Proposal Excellence (see Attachment 3, Evaluation Criteria).
M&IC	Multi-programmatic and Institutional Computing. Organization responsible for providing unclassified computing to all programs at the University Lawrence Livermore National Laboratory.
LLNL	Lawrence Livermore National Laboratory.

7.2 Hardware

b	bit. A single, indivisible binary unit of electronic information.
B	Byte. A collection of eight (8) bits.
32b floating-point arithmetic	Executable binaries (user applications) with 32b (4B) floating-point number representation and arithmetic. Note that this is independent of the number of bytes (4 our 8) utilized for memory reference addressing.

32b virtual memory addressing	All virtual memory addresses in a user application are 32b (4B) integers. Note that this is independent of the type of floating-point number representation and arithmetic.
64b floating-point arithmetic	Executable binaries (user applications) with 64b (8B) floating-point number representation and arithmetic. Note that this is independent of the number of bytes (4 our 8) utilized for memory reference addressing.
64b virtual memory addressing	All virtual memory addresses in a user application are 64b (8B) integers. Note that this is independent of the type of floating-point number representation and arithmetic. Note that all user applications should be compiled, loaded with Offeror supplied libraries and executed with 64b virtual memory addressing by default.
CE	On-site hardware customer engineer performing hardware maintenance with DOE Q-clearance.
Cluster	A set of SMPs connected via a scalable network technology. The network will support high bandwidth, low latency message passing. It will also support remote memory referencing.
CNSU	Compute Node Scalable Unit (CNSU) is the identical replicate unit for compute nodes. It contains four (4x42U) racks with eighty two (82x2U) compute nodes (2x21+2x20) and two (2x1.5U) Cisco management Ethernet switches ; one (1x47U) rack with fourteen (14x2U) compute nodes and one (1x17U, 128-way, 96D32U) Quadrics QsNet Elan3 switch.
CPU	Central Processing Unit or processor. A VLSI chip constituting the computational core (integer, floating point, and branch units), registers and memory interface (virtual memory translation, TLB and bus controller).
CWFS	Cluster Wide File System. The file system that is visible from every node in the system with scalable performance. This is a synonym for Lustre Lite and supporting I/O infrastructure hardware.
FLOP or OP	Floating Point OPeration.
FLOPS or OPS	Plural of FLOP.
FLOP/s or OP/s	Floating Point OPeration per second.
FRU	Field Replaceable Unit (FRU) is an aggregation of parts that is a single unit and can be replaced upon failure.
FSB	Front-side bus
FSU	Example First Scalable Unit (FSU) build for MCR contains six (6x42U racks with at least two (2x2U) management nodes (not on QsNet), at least two (2x4U) Login nodes, at least thirty two (32x2U) gateway nodes, at least two (2x4U) MDS (fail-over) nodes, at least sixty (60x2U) compute nodes (configured as a debug partition), at least ten (10x2U) hot spare compute nodes, three Cisco management Ethernet switches and the first (1x17U, 128-way, 96D32U) QsNet Elan3 switch.
GB	gigaByte. gigaByte is a billion base 10 bytes. This is typically used in every context except for Random Access Memory size and is 10^9 (or 1,000,000,000) bytes.

GiB	gibiByte. gibiByte is a billion base 2 bytes. This is typically used in terms of Random Access Memory and is 2^{30} (or 1,073,741,824) bytes. For a complete description of SI units for prefixing binary multiples see URL: `http://physics.nist.gov/cuu/Units/binary.html`
GFE	Government Furnished Equipment (GFE) is equipment supplied to the Offeror by the University when MCR build or installation takes place.
GFLOP/s or GOP/s	gigaFLOP/s. Billion (10^9 = 1,000,000,000) 64-bit floating point operations per second.
HSC	Hot Spare Cluster. A set of nodes on-site at LLNL that can be use as a hot spare pool constructed as a stand alone cluster. This HSC will be used to run diagnostics on failing nodes (after they are swapped out of MCR) to determine root cause for failures and to potentially test software releases.
MB	megaByte. megaByte is a million base 10 bytes. This is typically used in every context except for Random Access Memory size and is 10^6 (or 1,000,000) bytes.
MiB	mebiByte. mebiByte is a million base 2 bytes. This is typically used in terms of Random Access Memory and is 2^{20} (or 1,048,576) bytes. For a complete description of SI units for prefixing binary multiples see URL: `http://physics.nist.gov/cuu/Units/binary.html`
MFLOP/s or MOP/s	megaFLOP/s. Million (10^6 = 1,000,000) 64-bit floating point operations per second.
Mpixel	megapixel. Million (10^6 = 1,000,000) pixels.
Mpolygons	megapolygon. Million (10^6 = 1,000,000) polygon.
MTBF	Mean Time Between Failure. A measurement of the expected reliability of the system or component. The MTBF figure can be developed as the result of intensive testing, based on actual product experience, or predicted by analyzing known factors. See URL: `http://www.t-cubed.com/faq_mtbf.htm`
Node	Two or four Intel Pentium 4 microprocessors in an SMP configuration with the Linux operating system and possibly local disk.
Peak Rate	The maximum number of 64-bit floating-point instructions (add, subtract, multiply or divide) per second that could conceivably be retired by the system. For microprocessors the peak rate is typically calculated as the maximum number of floating point instructions retired per clock times the clock rate.
Pixel	The smallest image-forming unit of a video display.
Polygon	A closed plane figure bounded by three or more line segments. Aggregations of polygons in three-dimensional space are commonly used in computer visualization as a simplification to represent more complicated (smooth) three-dimensional shapes.
POST	Power-On Self Test (POST) is a set of diagnostics that run when the node is powered on to detect all hardware components and verify correct functioning.

RPC	Remote Power Control (RPC) is a rack mounted device (that may be combined with the SPC) that concentrates the power cords of nodes in the rack and allows for remote power management of the nodes via telnet over the management Ethernet.
SPC	Serial Port Concentrator (SPC) is a rack mounted device (that may be combined with the RPC) that connects the serial ports of nodes to the management Ehternet via reverse telnet protocol. This allows system administrators to log into the serial port of every node via the management network and perform management actions on the node. In addition, this interface allows the system administrators to set up telnet sessions with each node and log all console traffic.
Scalable	A system attribute that increases in performance or size as some function of the peak rating of the system. The scaling regime of interest is at least within the range of 1 teraFLOP/s to 60.0 (and possibly to 120.0) teraFLOP/s peak rate.
SMP	Shared memory Multi-Processor. A set of CPUs sharing random access memory within the same memory address space. The CPUs are connected via a high speed, low latency mechanism to the set of hierarchical memory components. The memory hierarchy consists of at least processor registers, cache and memory. The cache will also be hierarchical. If there are multiple caches, they will be kept coherent automatically by the hardware. The main memory will be UMA architecture. The access mechanism to every memory element will be the same from every processor. More specifically, all memory operations are done with load/store instructions issued by the CPU to move data to/from registers from/to the memory.
Tera-Scale	The environment required to fully support production-level, realized teraFLOP/s performance. This environment includes a robust and balanced processor, memory, mass storage, I/O, and communications subsystems; robust code development environment, tools and operating systems; and an integrated cluster wide systems management and full system reliability and availability.
TB	TeraByte. TeraByte is a trillion base 10 bytes. This is typically used in every context except for Random Access Memory size and is 10^{12} (or 1,000,000,000,000) bytes.
TiB	TebiByte. TebiByte is a trillion bytes base 2 bytes. This is typically used in terms of Random Access Memory and is 2^{40} (or 1,099,511,627,776) bytes. For a complete description of SI units for prefixing binary multiples see URL: `http://physics.nist.gov/cuu/Units/binary.html`
TFLOP/s	teraFLOP/s. Trillion (10^{12} = 1,000,000,000,000) 64-bit floating point operations per second.
UMA	Uniform Memory Access architecture. The distance in processor clocks between processor registers and every element of main memory is the same. That is, a load/store operation has the same latency, no matter where the target location is in main memory.

7.3 Software

32b executable	Executable binaries (user applications) with 32b (4B) virtual memory addressing. Note that this is independent of the number of bytes (4 our 8) utilized for floating-point number representation and arithmetic.
64b executable	Executable binaries (user applications) with 64b (8B) virtual memory addressing. Note that this is independent of the number of bytes (4 our 8) utilized for floating-point number representation and arithmetic. Note that all user applications should be compiled, loaded with Offeror supplied libraries and executed with 64b virtual memory addressing by default.
API (Application Programming Interface)	Syntax and semantics for invoking services from within an executing application. All APIs will be available to both Fortran and C programs, although implementation issues (such as whether the Fortran routines are simply wrappers for calling C routines) are up to the supplier.
BIOS	Basic Input-Output System (BIOS) is low level (typically assembly language) code usually held in flash memory on the node that tests and functions the hardware upon power-up or reset or reboot and loads the operating system.
Current standard	Term applied when an API is not "frozen" on a particular version of a standard, but will be upgraded automatically by Offeror as new specifications are released (e.g., "MPI version 2.0" refers to the standard in effect at the time of writing this document, while "current version of MPI" refers to further versions that take effect during the lifetime of this contract.
Fully supported (as applied to system software and tools)	A product-quality implementation, documented and maintained by the HPC machine supplier or an affiliated software supplier.
Gang Scheduling	When a user job is scheduled to run, the gang scheduler must contemporaneously allocate to CPUs all the threads and processes within that job (either within an SMP or within the cluster of SMPs). This scheduling capability must control all threads and processes within the SMP cluster environment.
GFS	Government Furnished Software (GFS) is software supplied to the Offeror by the University when MCR build or installation takes place.

Job	A job is a cluster wide abstraction similar to a POSIX session, with certain characteristics and attributes. Commands will be available to manipulate a job as a single entity (including kill, modify, query characteristics, and query state). The characteristics and attributes required for each session type are as follows: 1) interactive session: an interactive session will include all cluster wide processes executed as a child (whether direct or indirect through other processes) of a login shell and will include the login shell process as well. Normally, the login shell process will exist in a process chain as follows: init, inetd, [sshd | telnetd | rlogind | xterm | cron], then shell. 2) batch session: a batch session will include all cluster wide processes executed as a child (whether direct or indirect through other processes) of a shell process executed as a child process of a batch system shepherd process, and will include the batch system shepherd process as well. 3) ftp session: an ftp session will include an ftpd and all its child processes. 4) kernel session: all processes with a pid of 0. 5) idle session: this session does not necessarily actually consist of identifiable processes. It is a pseudo-session used to report the lack of use of resources. 6) system session: all processes owned by root that are not a part of any other session.
LinuxBIOS	An implementation of Linux stored in the node BIOS. This allows nodes to boot from BIOS flash ROM in less than thirty seconds. See www.linuxbios.org.
Lustre **Lustre Lite**	Lustre and Lustre Lite are cluster wide file systems based on object technology. See www.lustre.org for more details.
MPI	Message Passing Interface Version 1.2 or later. See, for example, http://www-unix.mcs.anl.gov/mpi/mpich/, or http://www.mpi-forum.org/docs/mpi-20-html/mpi2-report.html
Published (as applied to APIs):	Where an API is not required to be consistent across platforms, the capability lists it as "published," referring to the fact that it will be documented and supported, although it will be Offeror- or even platform-specific.
Single-point control (as applied to tool interfaces)	Refers to the ability to control or acquire information on all processes/PEs using a single command or operation.
Standard (as applied to APIs)	Where an API is required to be consistent across platforms, the reference standard is named as part of the capability. The implementation will include all routines defined by that standard (even if some simply result in no-ops on a given platform).
XXX-compatible (as applied to system software and tool definitions)	Requires that a capability be compatible, at the interface level, with the referenced standard, although the lower-level implementation details will differ substantially (e.g., "NFSv4-compatible" means that the distributed file system will be capable of handling standard NFSv4 requests, but need not conform to NFSv4 implementation specifics).

End of Section 7